Public-key-encryption data-communication system and data-communication-system forming method

Patent Number:	EP1130844
Publication date:	2001-09-05
Inventor(s):	ISHIBASHI YOSHIHITO (JP); MATSUYAMA SHINAKO (JP); KON MASASHI (JP); FUTAMARA ICHIRO (JP); WATANABE HIDEAKI (JP)
Applicant(s):	SONY CORP (JP)
Requested Patent:	JP2001320356
Application Number:	EP20010104908 20010228
Priority Number (s):	JP20000054091 20000229; JP20000123027 20000424
IPC Classification:	H04L9/32
EC Classification:	H04L9/32T
Equivalents:	US2001034834
Cited Documents:	

Abstract

A public-key-encryption data-communication system includes a public-key-certificate issuer authority. The public-key-certificate issuer authority performs the issuance of a public key certificate and management operations, certification of a subject to be certificated, which is a certificate issuing request, and management such as registration processing are executed by a root registration authority or each registration authority. The public-key-certificate issuer authority performs processing for validating, invalidating, and deleting the certificate in accordance with a request from the root registration authority. The root registration authority accepts a request for issuing a public key certificate corresponding to the subject to be certificated which is under the control of a certificated registration authority, and transfers it to the public-key-certificate issuer authority in a form in which a signature is added to it. Processes by the public-key-certificate issuer authority, the root registration authority, the registration authority are separated, whereby the need for new implementation of user recognition, certificate issuance, registration, and management is eliminated.

(19)日本国特許庁(JP)

(12)公開特許公報（A）

(11)特許出願公開番号

特開２００１－３２０３５６

（Ｐ２００１－３２０３５６Ａ）

(43)公開日　平成13年11月16日(2001.11.16)

(51)Int.Cl.⁷		識別記号	FI			テーマコード(参考)
Ｈ０４Ｌ	9/08		Ｇ０９Ｃ	1/00	６６０　Ｅ	5J104
Ｇ０９Ｃ	1/00	６６０	Ｈ０４Ｌ	9/00	６０１　Ｆ	
Ｈ０４Ｌ	9/32				６７５　Ｄ	

審査請求　未請求　請求項の数２９　ＯＬ　（全３６頁）

(21)出願番号　特願2000-123027(P2000-123027)

(22)出願日　平成12年4月24日(2000.4.24)

(31)優先権主張番号　特願2000-54091(P2000-54091)

(32)優先日　平成12年2月29日(2000.2.29)

(33)優先権主張国　日本（ＪＰ）

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川6丁目7番35号

(72)発明者　二村　一郎
東京都品川区北品川6丁目7番35号　ソニー株式会社内

(72)発明者　石橋　義人
東京都品川区北品川6丁目7番35号　ソニー株式会社内

(74)代理人　100101801
弁理士　山田　英治　（外2名）

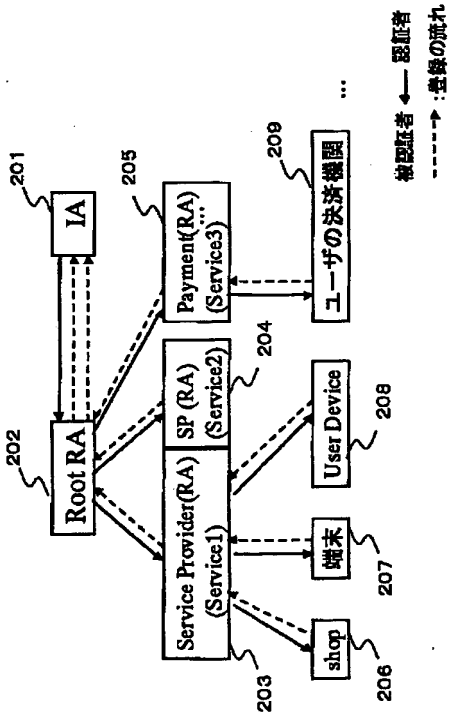
最終頁に続く

(54)【発明の名称】公開鍵系暗号を使用したデータ通信システムおよびデータ通信システム構築方法

(57)【要約】

【課題】　公開鍵証明書発行機能とユーザ登録、管理機能を分離することにより、効率的管理を実現する公開鍵系暗号を使用したデータ通信システムを提供する。

【解決手段】　公開鍵証明書発行局（ＩＡ）が公開鍵証明書の発行、管理業務を行ない、証明書の発行要求である認証対象の認証、登録処理等の管理はルート登録局（ルートＲＡ）または各登録局（ＲＡ）が実行する。公開鍵証明書発行局は、ルート登録局の要求に従って、証明書の有効、無効、削除処理を行なう。ルート登録局は認証した登録局の管轄する認証対象の公開鍵証明書発行要求を受理して署名を付して公開鍵証明書発行局に転送する。公開鍵証明書発行局（ＩＡ）、ルート登録局、登録局（ＲＡ）の処理が切り分けられ、サービス毎の新たなユーザ確認、証明書の発行、登録、管理が不要となる。



【特許請求の範囲】

【請求項１】公開鍵系暗号方式を使用したデータ転送を行なう認証対象の公開鍵証明書を発行する公開鍵証明書発行局と、

前記公開鍵証明書発行局と相互にデータ転送を実行するルート登録局であり、該ルート登録局の管轄する認証対象の認証処理を行なうとともに該認証対象の公開鍵証明書の発行要求を前記公開鍵証明書発行局に対して実行するルート登録局と、

前記ルート登録局と相互にデータ転送を実行する登録局であり、該登録局の管轄する認証対象の認証処理を行なうとともに該認証対象の公開鍵証明書の発行要求を前記ルート登録局に対して実行する登録局と、

を有することを特徴とする公開鍵系暗号を使用したデータ通信システム。

【請求項２】前記ルート登録局は、複数の登録局を認証対象とし、

前記複数の登録局の各々は管轄下のサービスプロバイダまたはユーザ端末またはユーザのいずれかを認証対象とする構成であることを特徴とする請求項１に記載の公開鍵系暗号を使用したデータ通信システム。

【請求項３】前記登録局または前記登録局の管轄下のサービスプロバイダは、

前記登録局または前記登録局の管轄下のサービスプロバイダの管轄する認証対象の１つの公開鍵証明書を複数の異なるサービスに適用する構成としたことを特徴とする請求項１に記載の公開鍵系暗号を使用したデータ通信システム。

【請求項４】前記ルート登録局は、管轄する認証対象の複数の登録局の１つとして決済処理を実行するクリアリングセンタを有し、

前記ルート登録局の管轄する前記クリアリングセンタ以外の登録局または該登録局管轄下のサービスプロバイダの提供するサービスに関する決済処理を前記クリアリングセンタを介して発行された公開鍵証明書を用いた処理において実行する構成であることを特徴とする請求項１に記載の公開鍵系暗号を使用したデータ通信システム。

【請求項５】前記公開鍵証明書発行局は、公開鍵証明書を発行した認証対象識別子と公開鍵および公開鍵証明書の対応リストを保持し、

前記ルート登録局または前記登録局のいずれかは、公開鍵証明書を発行した認証対象の認証用データを含む認証対象毎のエンティテイ・データを保持する構成であることを特徴とする請求項１に記載の公開鍵系暗号を使用したデータ通信システム。

【請求項６】前記公開鍵証明書は、前記公開鍵証明書発行局の電子署名フィールドを有し、

前記電子署名フィールドに生成される電子署名の署名アルゴリズムは、複数のアルゴリズムが適用可能であり、前記公開鍵証明書には適用アルゴリズムを識別するフィールドが設けられた構成であることを特徴とする請求項１に記載の公開鍵系暗号を使用したデータ通信システム。

【請求項７】前記公開鍵証明書発行局と前記ルート登録局とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行する構成であり、

前記ルート登録局と前記登録局とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行する構成であり、

前記登録局と前記認証対処とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行する構成であることを特徴とする請求項１に記載の公開鍵系暗号を使用したデータ通信システム。

【請求項８】前記公開鍵証明書発行局、前記ルート登録局、前記登録局、および前記認証対象いずれか２者間において転送されるデータにはデータ送信側の電子署名を生成して転送する構成であることを特徴とする請求項１に記載の公開鍵系暗号を使用したデータ通信システム。

【請求項９】前記ルート登録局または登録局の少なくともいずれかは、管轄下の認証対象の公開鍵証明書に関する失効リストを保有して、該失効リストの更新処理を実行するともに、該更新処理に対応するデータの処理要求を前記公開鍵証明書発行局に行なう構成であることを特徴とする請求項１に記載の公開鍵系暗号を使用したデータ通信システム。

【請求項１０】前記ルート登録局または登録局の少なくともいずれかは、該ルート登録局または登録局管轄下の複数のサービスの各々に対する複数の公開鍵証明書の発行要求を行なう構成であることを特徴とする請求項１に記載の公開鍵系暗号を使用したデータ通信システム。

【請求項１１】前記公開鍵証明書は、該公開鍵証明書を発行した公開鍵証明書発行局の共通の電子署名がなされ、前記公開鍵証明書発行局の発行した１つの公開鍵証明書の検証処理の可能なルート登録局、登録局、サービスプロバイタ、またはユーザデバイスは、同一の公開鍵証明書発行局の発行した異なる公開鍵証明書の検証処理をオフラインで実行可能とした構成であることを特徴とする請求項１に記載の公開鍵系暗号を使用したデータ通信システム。

【請求項１２】前記登録局は、

ユーザ端末によって利用可能なコンテンツまたはサービスの提供を可能とするコンテンツまたはサービスの流通インフラを提供または管理する機関であるシステムホルダとして構成され、

前記システムホルダは、サービスプロバイダおよびユーザ端末を管轄し、認証対象とした構成であることを特徴とする請求項１に記載の公開鍵系暗号を使用したデータ通信システム。

【請求項１３】前記ルート登録局は、
異なるコンテンツまたはサービスの流通インフラを提供または管理する異なる複数のシステムホルダを管轄下に配し、システムホルダの管轄下のサービスプロバイダおよびユーザ端末からのシステムホルダを介する公開鍵証明書発行要求を受領して、前記公開鍵証明書発行局に対する公開鍵証明書発行要求を実行する構成であることを特徴とする請求項１２に記載の公開鍵系暗号を使用したデータ通信システム。
【請求項１４】前記システムホルダの管轄下には、
該システムホルダの提供または管理するコンテンツまたはサービスの流通インフラを利用してコンテンツ提供を行なうコンテンツクリエイタを有し、
該システムホルダは、前記コンテンツクリエイタを認証対象とした構成であることを特徴とする請求項１２に記載の公開鍵系暗号を使用したデータ通信システム。
【請求項１５】共通の公開鍵証明書発行局の管轄する複数の異なるシステムホルダの管轄下にあるユーザデバイスは、共通の公開鍵証明書発行局の公開鍵を有する構成であることを特徴とする請求項１２に記載の公開鍵系暗号を使用したデータ通信システム。
【請求項１６】公開鍵系暗号を使用したデータ通信システム構築方法において、
認証対象から登録局に対して公開鍵証明書の発行を要求するステップと、
前記登録局から該登録局を認証したルート登録局に対して前記認証対象からの公開鍵証明書発行要求を転送するステップと、
前記ルート登録局から該ルート登録局を認証した公開鍵証明書発行局に対して前記認証対象からの公開鍵証明書発行要求を転送するステップと、
を有することを特徴とする公開鍵系暗号を使用したデータ通信システム構築方法。
【請求項１７】前記ルート登録局は、複数の登録局を認証対象とし、
前記複数の登録局の各々は管轄下のサービスプロバイダまたはユーザ端末またはユーザのいずれかを認証対象とすることを特徴とする請求項１６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。
【請求項１８】前記登録局または前記登録局の管轄下のサービスプロバイダは、
前記登録局または前記登録局の管轄下のサービスプロバイダの管轄する認証対象の１つの公開鍵証明書を複数の異なるサービスに適用することを特徴とする請求項１６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。
【請求項１９】前記ルート登録局は、管轄する認証対象の複数の登録局の１つとして決済処理を実行するクリアリングセンタを有し、
前記ルート登録局の管轄する前記クリアリングセンタ以外の登録局または該登録局管轄下のサービスプロバイダの提供するサービスに関する決済処理を前記クリアリングセンタを介して発行された公開鍵証明書を用いた処理において実行することを特徴とする請求項１６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。
【請求項２０】前記公開鍵証明書発行局は、公開鍵証明書を発行した認証対象識別子と公開鍵および公開鍵証明書の対応リストを保持し、
前記ルート登録局または前記登録局のいずれかは、公開鍵証明書を発行した認証対象の認証用データを含む認証対象毎のエンティテイ・データを保持することを特徴とする請求項１６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。
【請求項２１】前記公開鍵証明書発行局と前記ルート登録局とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行し、
前記ルート登録局と前記登録局とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行し、
前記登録局と前記認証対処とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行することを特徴とする請求項１６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。
【請求項２２】前記公開鍵証明書発行局、前記ルート登録局、前記登録局、および前記認証対象いずれか２者間において転送されるデータにはデータ送信側の電子署名を生成して転送することを特徴とする請求項１６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。
【請求項２３】前記ルート登録局または登録局の少なくともいずれかは、管轄下の認証対象の公開鍵証明書に関する失効リストを保有して、該失効リストの更新処理を実行するともに、該更新処理に対応するデータの処理要求を前記公開鍵証明書発行局に行なうことを特徴とする請求項１６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。
【請求項２４】前記ルート登録局または登録局の少なくともいずれかは、該ルート登録局または登録局管轄下の複数のサービスの各々に対する複数の公開鍵証明書の発行要求を行なう構成であることを特徴とする請求項１６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。
【請求項２５】前記公開鍵証明書は、該公開鍵証明書を発行した公開鍵証明書発行局の共通の電子署名がなされ、前記公開鍵証明書発行局の発行した１つの公開鍵証明書の検証処理の可能なルート登録局、登録局、サービスプロバイタ、またはユーザデバイスは、同一の公開鍵証明書発行局の発行した異なる公開鍵証明書の検証処理

をオフラインで実行可能とした構成であることを特徴とする請求項１６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。

【請求項２６】前記登録局は、
ユーザ端末によって利用可能なコンテンツまたはサービスの提供を可能とするコンテンツまたはサービスの流通インフラを提供または管理する機関であるシステムホルダとして構成され、
前記システムホルダは、サービスプロバイダおよびユーザ端末を管轄し、認証処理を実行することを特徴とする請求項１６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。

【請求項２７】前記ルート登録局は、
異なるコンテンツまたはサービスの流通インフラを提供または管理する異なる複数のシステムホルダを管轄下に配し、システムホルダの管轄下のサービスプロバイダおよびユーザ端末からのシステムホルダを介する公開鍵証明書発行要求を受領し、
前記公開鍵証明書発行局に対する公開鍵証明書発行要求を実行することを特徴とする請求項２６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。

【請求項２８】前記システムホルダの管轄下には、
該システムホルダの提供または管理するコンテンツまたはサービスの流通インフラを利用してコンテンツ提供を行なうコンテンツクリエイタを有し、
該システムホルダは、前記コンテンツクリエイタを認証処理を実行することを特徴とする請求項２６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。

【請求項２９】共通の公開鍵証明書発行局の管轄する複数の異なるシステムホルダの管轄下にあるユーザデバイスは、共通の公開鍵証明書発行局の公開鍵を有し、
ユーザデバイス間において、公開鍵証明書発行局の公開鍵を使用した相互認証処理を実行する構成であることを特徴とする請求項２６に記載の公開鍵系暗号を使用したデータ通信システム構築方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は電子配信システムにおいて暗号化データ送信に使用される公開鍵の正当性を証明するための公開鍵証明書発行システムおよびデータ通信方法に関する。さらに、詳細には、データ送信サービスを行なうエンティテイが、公開鍵証明書を発行する認証局の機能を構築することなく、汎用的に公開鍵、公開鍵証明書を使用することを可能とする公開鍵証明書発行システムおよびデータ通信方法に関する。

【０００２】

【従来の技術】昨今、ゲームプログラム、音声データ、画像データ、文書作成プログラム等、様々なソフトウエアデータ（以下、これらをコンテンツ（Content）と呼ぶ）が、インターネット等のネットワークを介して流通している。また、オンラインショッピング等、ネットワークを介した商品売買も次第に盛んになってきている。

【０００３】このようなネットワークを介したデータ通信においては、データ送信側とデータ受信側とが互いに正規なデータ送受信対象であることを確認した上で、必要な情報を転送する、すなわちセキュリティを考慮したデータ転送構成をとるのが一般的となっている。データ転送の際のセキュリティ構成を実現する１つの手法が、転送データの暗号化処理、データに対する署名処理である。

【０００４】暗号化データは、所定の手続きによる復号化処理によって利用可能な復号データ（平文）に戻すことができる。このような情報の暗号化処理に暗号化鍵を用い、復号化処理に復号化鍵を用いるデータ暗号化、復号化方法は従来からよく知られている。

【０００５】暗号化鍵と復号化鍵を用いるデータ暗号化・復号化方法の態様には様々な種類あるが、その１つの例としていわゆる公開鍵暗号方式と呼ばれる方式がある。公開鍵暗号方式は、発信者と受信者の鍵を異なるものとして、一方の鍵を不特定のユーザが使用可能な公開鍵として、他方を秘密に保つ秘密鍵とするものである。例えば、データ暗号化鍵を公開鍵とし、復号鍵を秘密鍵とする。あるいは、認証子生成鍵を秘密鍵とし、認証子復号鍵を公開鍵とする等の態様において使用される。

【０００６】暗号化、復号化に共通の鍵を用いるいわゆる共通鍵暗号化方式と異なり、公開鍵暗号方式では秘密に保つ必要のある秘密鍵は、特定の１人が持てばよいため鍵の管理において有利である。ただし、公開鍵暗号方式は共通鍵暗号化方式に比較してデータ処理速度が遅く、秘密鍵の配送、ディジタル署名等のデータ量の少ない対象に多く用いられている。公開鍵暗号方式の代表的なものにはＲＳＡ（Rivest-Shamir-Adleman）暗号がある。これは非常に大きな２つの素数（例えば１５０桁）の積を用いるものであり、大きな２つの素数（例えば１５０桁）の積の素因数分解する処理の困難さを利用している。

【０００７】公開鍵暗号方式では、不特定多数に公開鍵を使用可能とする構成であり、配布する公開鍵が正当なものであるか否かを証明する証明書、いわゆる公開鍵証明書を使用する方法が多く用いられている。例えば、利用者Ａが公開鍵、秘密鍵のペアを生成して、生成した公開鍵を認証局に対して送付して公開鍵証明書を認証局から入手する。利用者Ａは公開鍵証明書を一般に公開する。不特定のユーザは公開鍵証明書から所定の手続きを経て公開鍵を入手して文書等を暗号化して利用者Ａに送付する。利用者Ａは秘密鍵を用いて暗号化文書等を復号する等のシステムである。また、利用者Ａは、秘密鍵を用いて文書等に署名を付け、不特定のユーザが公開鍵証明書から所定の手続きを経て公開鍵を入手して、その署名の検証を行なうシステムである。

【０００８】公開鍵証明書について図１を用いて説明する。公開鍵証明書は、公開鍵暗号方式における認証局（ＣＡ：Certificate AuthorityまたはＩＡ：Issuer Authority）が発行する証明書であり、ユーザが自己のＩＤ、公開鍵等を認証局に提出することにより、認証局側が認証局のＩＤや有効期限等の情報を付加し、さらに認証局による署名を付加して作成される証明書である。

【０００９】図１に示す公開鍵証明書は、証明書のバージョン番号、認証局（ＩＡ）が証明書利用者に対し割り付ける証明書の通し番号、電子署名に用いたアルゴリズムおよびパラメータ、認証局の名前、証明書の有効期限、証明書利用者の名前（ユーザＩＤ）、証明書利用者の公開鍵並びに電子署名を含む。

【００１０】電子署名は、証明書のバージョン番号、認証局が証明書利用者に対し割り付ける証明書の通し番号、電子署名に用いたアルゴリズムおよびパラメータ、認証局の名前、証明書の有効期限、証明書利用者の名前並びに証明書利用者の公開鍵全体に対しハッシュ関数を適用してハッシュ値を生成し、そのハッシュ値に対して認証局の秘密鍵を用いて生成したデータである。

【００１１】認証局は、図１に示す公開鍵証明書を発行するとともに、有効期限が切れた公開鍵証明書を更新し、不正を行った利用者の排斥を行うための不正者リストの作成、管理、配布（これをリボケーション：Revocationと呼ぶ）を行う。また、必要に応じて公開鍵・秘密鍵の生成も行う。

【００１２】一方、この公開鍵証明書を利用する際には、利用者は自己が保持する認証局の公開鍵を用い、当該公開鍵証明書の電子署名を検証し、電子署名の検証に成功した後に公開鍵証明書から公開鍵を取り出し、当該公開鍵を利用する。従って、公開鍵証明書を利用する全ての利用者は、共通の認証局の公開鍵を保持している必要がある。

【００１３】

【発明が解決しようとする課題】上述のような認証局発行の公開鍵証明書を用いた公開鍵暗号方式によるデータ送信システムにおいては、使用する公開鍵が異なれば、その公開鍵に対して新たに認証局に対して公開鍵証明書の発行を依頼、あるいは認証局構成を持つ認証システムを構築することが必要となる。すなわち、例えばコンテンツの配信、商品提供サービスを行なうサービスプロバイダは、新たなサービス（新たな電子配信システム）を開始し、新たな公開鍵の使用を開始する際に、逐一、新たな公開鍵に対応する公開鍵証明書の発行、管理を認証局に依頼、あるいは認証局構成を持つ認証システムを構築しなければならず、多大なコスト、時間を要するという問題があった。また、異なる認証局間で発行された証明書を用いてやりとりする場合は、その証明書中の発行局の署名を検証するために必ずセンタに接続して発行局の署名検証鍵を入手する必要があり、オフラインでの利用には適さなかった。

【００１４】本発明は、このような公開鍵証明書の発行システムを簡易化し、サービスプロバイダが新たなサービスを開始する際の公開鍵証明書の使用を容易にすることを可能とした公開鍵系暗号を使用したデータ通信システムおよびデータ通信方法を提供することを目的とする。

【００１５】

【課題を解決するための手段】本発明の第１の側面は、公開鍵系暗号方式を使用したデータ転送を行なう認証対象の公開鍵証明書を発行する公開鍵証明書発行局と、前記公開鍵証明書発行局と相互にデータ転送を実行するルート登録局であり、該ルート登録局の管轄する認証対象の認証処理を行なうとともに該認証対象の公開鍵証明書の発行要求を前記公開鍵証明書発行局に対して実行するルート登録局と、前記ルート登録局と相互にデータ転送を実行する登録局であり、該登録局の管轄する認証対象の認証処理を行なうとともに該認証対象の公開鍵証明書の発行要求を前記ルート登録局に対して実行する登録局と、を有することを特徴とする公開鍵系暗号を使用したデータ通信システムにある。

【００１６】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記ルート登録局は、複数の登録局を認証対象とし、前記複数の登録局の各々は管轄下のサービスプロバイダまたはユーザ端末またはユーザのいずれかを認証対象とする構成であることを特徴とする。

【００１７】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記登録局または前記登録局の管轄下のサービスプロバイダは、前記登録局または前記登録局の管轄下のサービスプロバイダの管轄する認証対象の１つの公開鍵証明書を複数の異なるサービスに適用する構成としたことを特徴とする。

【００１８】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記ルート登録局は、管轄する認証対象の複数の登録局の１つとして決済処理を実行するクリアリングセンタを有し、前記ルート登録局の管轄する前記クリアリングセンタ以外の登録局または該登録局管轄下のサービスプロバイダの提供するサービスに関する決済処理を前記クリアリングセンタを介して発行された公開鍵証明書を用いた処理において実行する構成であることを特徴とする。

【００１９】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記公開鍵証明書発行局は、公開鍵証明書を発行した認証対象識別子と公開鍵および公開鍵証明書の対応リストを保持し、前記ルート登録局または前記登録局のいずれかは、公開鍵証明書を発行した認証対象の認証用データを含む認証対象毎のエンティティ・データを保持する構成であることを特徴とする。

【００２０】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記公開鍵証明書は、前記公開鍵証明書発行局の電子署名フィールドを有し、前記電子署名フィールドに生成される電子署名の署名アルゴリズムは、複数のアルゴリズムが適用可能であり、前記公開鍵証明書には適用アルゴリズムを識別するフィールドが設けられた構成であることを特徴とする。

【００２１】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記公開鍵証明書発行局と前記ルート登録局とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行する構成であり、前記ルート登録局と前記登録局とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行する構成であり、前記登録局と前記認証対処とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行する構成であることを特徴とする。

【００２２】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記公開鍵証明書発行局、前記ルート登録局、前記登録局、および前記認証対象いずれか２者間において転送されるデータにはデータ送信側の電子署名を生成して転送する構成であることを特徴とする。

【００２３】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記ルート登録局または登録局の少なくともいずれかは、管轄下の認証対象の公開鍵証明書に関する失効リストを保有して、該失効リストの更新処理を実行するとともに、該更新処理に対応するデータの処理要求を前記公開鍵証明書発行局に行なう構成であることを特徴とする。

【００２４】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記ルート登録局または登録局の少なくともいずれかは、該ルート登録局または登録局管轄下の複数のサービスの各々に対する複数の公開鍵証明書の発行要求を行なう構成であることを特徴とする。

【００２５】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記公開鍵証明書は、該公開鍵証明書を発行した公開鍵証明書発行局の共通の電子署名がなされ、前記公開鍵証明書発行局の発行した１つの公開鍵証明書の検証処理の可能なルート登録局、登録局、サービスプロバイタ、またはユーザデバイスは、同一の公開鍵証明書発行局の発行した異なる公開鍵証明書の検証処理をオフラインで実行可能とした構成であることを特徴とする。

【００２６】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記登録局は、ユーザ端末によって利用可能なコンテンツまたはサービスの提供を可能とするコンテンツまたはサービスの流通インフラを提供または管理する機関であるシステムホルダとして構成され、前記システムホルダは、サービスプロバイダおよびユーザ端末を管轄し、認証対象とした構成であることを特徴とする。

【００２７】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記ルート登録局は、異なるコンテンツまたはサービスの流通インフラを提供または管理する異なる複数のシステムホルダを管轄下に配し、システムホルダの管轄下のサービスプロバイダおよびユーザ端末からのシステムホルダを介する公開鍵証明書発行要求を受領して、前記公開鍵証明書発行局に対する公開鍵証明書発行要求を実行する構成であることを特徴とする。

【００２８】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、前記システムホルダの管轄下には、該システムホルダの提供または管理するコンテンツまたはサービスの流通インフラを利用してコンテンツ提供を行なうコンテンツクリエイタを有し、該システムホルダは、前記コンテンツクリエイタを認証対象とした構成であることを特徴とする。

【００２９】さらに、本発明の公開鍵系暗号を使用したデータ通信システムにおいて、共通の公開鍵証明書発行局の管轄する複数の異なるシステムホルダの管轄下にあるユーザデバイスは、共通の公開鍵証明書発行局の公開鍵を有する構成であることを特徴とする。

【００３０】さらに、本発明の第２の側面は、公開鍵系暗号を使用したデータ通信システム構築方法において、認証対象から登録局に対して公開鍵証明書の発行を要求するステップと、前記登録局から該登録局を認証したルート登録局に対して前記認証対象からの公開鍵証明書発行要求を転送するステップと、前記ルート登録局から該ルート登録局を認証した公開鍵証明書発行局に対して前記認証対象からの公開鍵証明書発行要求を転送するステップと、を有することを特徴とする公開鍵系暗号を使用したデータ通信システム構築方法にある。

【００３１】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記ルート登録局は、複数の登録局を認証対象とし、前記複数の登録局の各々は管轄下のサービスプロバイダまたはユーザ端末またはユーザのいずれかを認証対象とすることを特徴とする。

【００３２】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記登録局または前記登録局の管轄下のサービスプロバイダは、前記登録局または前記登録局の管轄下のサービスプロバイダの管轄する認証対象の１つの公開鍵証明書を複数の異なるサービスに適用することを特徴とする。

【００３３】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記ルート登録局は、管轄する認証対象の複数の登録局の１つとして決済処理を実行するクリアリングセンタを有し、前記ルート登録局の管

轄する前記クリアリングセンタ以外の登録局または該登録局管轄下のサービスプロバイダの提供するサービスに関する決済処理を前記クリアリングセンタを介して発行された公開鍵証明書を用いた処理において実行することを特徴とする。

【００３４】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記公開鍵証明書発行局は、公開鍵証明書を発行した認証対象識別子と公開鍵および公開鍵証明書の対応リストを保持し、前記ルート登録局または前記登録局のいずれかは、公開鍵証明書を発行した認証対象の認証用データを含む認証対象毎のエンティテイ・データを保持することを特徴とする。

【００３５】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記公開鍵証明書発行局と前記ルート登録局とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行し、前記ルート登録局と前記登録局とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行し、前記登録局と前記認証対処とのデータ転送においては、相互認証処理を行い、相互認証が成立した場合において相互間のデータ転送を実行することを特徴とする。

【００３６】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記公開鍵証明書発行局、前記ルート登録局、前記登録局、および前記認証対象いずれか２者間において転送されるデータにはデータ送信側の電子署名を生成して転送することを特徴とする。

【００３７】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記ルート登録局または登録局の少なくともいずれかは、管轄下の認証対象の公開鍵証明書に関する失効リストを保有して、該失効リストの更新処理を実行するともに、該更新処理に対応するデータの処理要求を前記公開鍵証明書発行局に行なうことを特徴とする。

【００３８】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記ルート登録局または登録局の少なくともいずれかは、該ルート登録局または登録局管轄下の複数のサービスの各々に対する複数の公開鍵証明書の発行要求を行なう構成であることを特徴とする。

【００３９】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記公開鍵証明書は、該公開鍵証明書を発行した公開鍵証明書発行局の共通の電子署名がなされ、前記公開鍵証明書発行局の発行した１つの公開鍵証明書の検証処理の可能なルート登録局、登録局、サービスプロバイタ、またはユーザデバイスは、同一の公開鍵証明書発行局の発行した異なる公開鍵証明書の検証処理をオフラインで実行可能とした構成であることを特徴とする。

【００４０】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記登録局は、ユーザ端末によって利用可能なコンテンツまたはサービスの提供を可能とするコンテンツまたはサービスの流通インフラを提供または管理する機関であるシステムホルダとして構成され、前記システムホルダは、サービスプロバイダおよびユーザ端末を管轄し、認証処理を実行することを特徴とする。

【００４１】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記ルート登録局は、異なるコンテンツまたはサービスの流通インフラを提供または管理する異なる複数のシステムホルダを管轄下に配し、システムホルダの管轄下のサービスプロバイダおよびユーザ端末からのシステムホルダを介する公開鍵証明書発行要求を受領し、前記公開鍵証明書発行局に対する公開鍵証明書発行要求を実行することを特徴とする。

【００４２】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、前記システムホルダの管轄下には、該システムホルダの提供または管理するコンテンツまたはサービスの流通インフラを利用してコンテンツ提供を行なうコンテンツクリエイタを有し、該システムホルダは、前記コンテンツクリエイタを認証処理を実行することを特徴とする。

【００４３】さらに、本発明の公開鍵系暗号を使用したデータ通信方法において、共通の公開鍵証明書発行局の管轄する複数の異なるシステムホルダの管轄下にあるユーザデバイスは、共通の公開鍵証明書発行局の公開鍵を有し、ユーザデバイス間において、公開鍵証明書発行局の公開鍵を使用した相互認証処理を実行する構成であることを特徴とする。

【００４４】本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【００４５】

【発明の実施の形態】以下、図面を参照しながら、本発明の実施の形態について詳細に説明する。

【００４６】

【実施例】図２に本発明の公開鍵系暗号を使用したデータ通信システムおよびデータ通信方法の概要を説明する図を示す。

【００４７】図２において、ショップ２０６、端末２０７、ユーザデバイス２０８、ユーザの決済機関２０９が認証対象者、すなわち公開鍵暗号化方式によるデータ送受信を実行する主体となる。図２では、代表的な認証対象者としてショップ２０６、端末２０７、ユーザデバイス２０８、ユーザの決済機関２０９をそれぞれ１つづつ示しているが、これらは一般に多数存在し、また、これら以外にも様々な種類の認証対象者が存在することができる。

【００４８】各々の登録局（ＲＡ）の管轄下にある認証対象者のショップ２０６、端末２０７、ユーザデバイス

２０８、ユーザの決済機関２０９は、登録局（サービスプロバイダＲＡ）２０３、２０４、登録局（ペイメントＲＡ）２０５に対して、自己の使用する公開鍵に対応する公開鍵証明書の発行を要求する。

【００４９】登録局（ＲＡ：Registration Authority）２０３、２０４、２０５は、各サービスにおける対象（サービスに参加するエンティティ、機器）を認証、あるいはそのサービスへの参加者の支払者を認証する（支払に対する保証）。また、登録局２０３、２０４、２０５は、各サービスにおける対象（サービスに参加するエンティティ、機器、ユーザ）の使用する公開鍵の公開鍵証明書発行要求を受領し、これをルート登録局（ルートＲＡ）２０２を介して公開鍵証明書発行局（ＩＡ）２０１に転送する。ルート登録局（ルートＲＡ）２０２は、認証済みの登録局２０３、２０４、２０５からの公開鍵証明書発行要求を受理する。すなわち、ルート登録局（ルートＲＡ）２０２が公開鍵証明書発行要求を受領するのは、ルート登録局（ルートＲＡ）２０２によって認証された登録局からの要求のみである。

【００５０】図２において、例えば登録局（サービスプロバイダＲＡ）２０３，２０４は音楽データ、画像データ、ゲームプログラム等のコンテンツ配信のサービス提供を実行するサービスプロバイダであり、登録局（ペイメントＲＡ）２０５は、銀行等のユーザの決済機関２０９とデータ送受信を行ない、ユーザの電子マネーの決済処理を実行するクリアリングセンタである。これら、登録局（ＲＡ）についても図２に示すものは一例であり、この他にも様々なサービスを提供する各種の登録局（ＲＡ）が存在可能である。

【００５１】登録局（ＲＡ）は各サービス（システム）毎に存在し、その登録局（ＲＡ）を統括して認証するものとしてルートＲＡ（Root Registration Authority）２０２が存在する。ルートＲＡ（Root RA）２０２は次に述べるＩＡによって認証される。登録局（ＲＡ）２０３、２０４、２０５は、小規模なサービス主体であり、サービス提供者が独自のＲＡを持たない場合にはルートＲＡ（Root RA）２０２が機能を代行する事ができる。

【００５２】図２に示すＩＡ，２０１は公開鍵証明書発行局（ＩＡ：Issuer Authority）である。ルート登録局（ルートＲＡ）２０２、または登録局（ＲＡ）２０３～２０５との間で相互認証を行い、ルートＲＡ２０２、または登録局（ＲＡ）２０３～２０５から渡される公開鍵証明書発行要求主体である対象を識別する対象識別子（ＩＤ）、対象の公開鍵、その他の公開鍵証明書に書き込む情報を元に公開鍵証明書を作成して登録局（ＲＡ）２０３～２０５に配布する。

【００５３】公開鍵証明書発行局（ＩＡ）２０１に対して証明書発行を要求するルート登録局（ルートＲＡ）２０２、または登録局（ＲＡ）２０３～２０５は、公開鍵証明書発行局から認証されていることが条件となる。

【００５４】また、公開鍵証明書発行局（ＩＡ）２０１は、ルート登録局（ルートＲＡ）２０２、または登録局（ＲＡ）２０３～２０５の要求を受けて、公開鍵証明書の更新、無効化、削除あるいは対象者からの有効性確認に対する応答処理を行う。この公開鍵証明書発行局（ＩＡ：Issuer Authority）２０１は適切な法的機関の認定を受ける位置づけのものであり、その認可を持って認証されているものとする。

【００５５】図３および図４に公開鍵証明書発行局（ＩＡ：Issuer Authority）２０１と、ルート登録局（ルートＲＡ）２０２または登録局（ＲＡ）２０３～２０５、およびショップ２０６、端末２０７、ユーザデバイス２０８、ユーザの決済機関２０９等の認証対象者における処理を説明する図を示す。

【００５６】図３は、ショップ２０６、端末２０７、ユーザデバイス２０８、ユーザの決済機関２０９等の認証対象者自身が公開鍵暗号化方式に適用する公開鍵、秘密鍵を生成する場合の例、図４は、ルートＲＡ２０２または登録局（ＲＡ）２０３～２０５が公開鍵、秘密鍵を生成する場合の例である。なお、図３、図４に示すサービス提供者３０４は、ＲＡの機能を持たないサービスプロバイダである。

【００５７】図３に示す各処理について説明する。認証対象者３０３は、公開鍵暗号化方式に適用する公開鍵、秘密鍵を生成して、公開鍵の証明書発行要求を登録局（ＲＡ）３０２に対して実行する。この際、認証対象者３０３は、自己のＩＤと公開鍵を送信する。自己のＩＤは例えばユーザ自身の識別子、ユーザ端末の識別子等である。これらの情報を受領した登録局（ＲＡ）３０２は、認証対象の確認を実行した後、受領した認証対象のＩＤと公開鍵を公開鍵証明書発行局（ＩＡ）３０１に転送する。公開鍵証明書発行局（ＩＡ）３０１は、受領した認証対象のＩＤと公開鍵、その他の公開鍵証明書に書き込む情報を元に公開鍵証明書を作成して登録局（ＲＡ）またはルートＲＡを介して登録局（ＲＡ）３０２に証明書を配布する。登録局（ＲＡ）３０２は、認証対象者３０１に対して公開鍵証明書を転送する。

【００５８】さらに、認証対象者３０３が新たな公開鍵、秘密鍵を生成して更新処理を実行する場合は、新たに生成した公開鍵を自己のＩＤとともに登録局（ＲＡ）３０２に送付し、登録局（ＲＡ）３０２は、認証対象の確認を実行した後、受領した認証対象のＩＤと公開鍵を公開鍵証明書発行局（ＩＡ）３０１に転送して、公開鍵証明書発行局（ＩＡ）３０１は、受領した認証対象のＩＤと公開鍵、その他の公開鍵証明書に書き込む情報を元に新たな公開鍵証明書を作成して登録局（ＲＡ）またはルートＲＡを介して認証対象者に送信する。

【００５９】図３に示すように、登録局（ＲＡ）またはルート登録局（ルートＲＡ）３０２は、認証対象者の確認処理、機器情報、ユーザ情報の保持を行なうとともに

に、公開鍵証明書発行局（ＩＡ）３０１において発行された失効管理を行なう。

【００６０】公開鍵証明書発行局（ＩＡ）３０１の処理は、発行した公開鍵証明書に関する公開鍵と認証対象者のＩＤの管理、公開鍵証明書の発行処理、発行済み公開鍵証明書の無効化処理、発行した公開鍵証明書の有効性チェックを行なう。

【００６１】失効手続きとしては、公開鍵証明書発行局（ＩＡ）３０１が、登録局（ＲＡ）またはルート登録局（ルートＲＡ）３０２からの要求に基づいて、発行した公開鍵証明書の無効化処理を実行する。さらに、登録局（ＲＡ）またはルート登録局（ルートＲＡ）３０２が、失効通知を認証対象者３０３および失効通知を必要とするサービス提供者３０４に対して行なう。失効通知は、後段で説明するが、失効リストを管理する登録局（ＲＡ）またはルート登録局（ルートＲＡ）３０２が失効リストから失効者データを抜き出した差分データとして提供することができる。

【００６２】認証対象者３０３は、自己の公開鍵が使用可能、すなわち公開鍵証明書が有効であるか否かのチェックを公開鍵証明書発行局（ＩＡ）３０１に依頼することが可能であり、この場合、認証対象者３０３は自己のＩＤと公開鍵を公開鍵証明書発行局（ＩＡ）３０１に送付し、公開鍵証明書発行局（ＩＡ）３０１が管理する公開鍵と認証対象者のＩＤに基づいて有効性確認を行なうことができる。

【００６３】図４は、登録局（ＲＡ）またはルートＲＡ３０２が公開鍵、秘密鍵を生成する場合の例である。図４において、登録局（ＲＡ）またはルートＲＡ３０２が生成した公開鍵、秘密鍵は、認証対象者３０３に送付され、認証対象者３０３がこれを格納する。以下の処理は、図３の処理と同様である。

【００６４】なお、図３、図４において公開鍵証明書の失効管理は、登録局（ＲＡ）またはルートＲＡ３０２が行なう構成としてあるが、公開鍵証明書発行局（ＩＡ）３０１が失効管理を行なうように構成してもよい。

【００６５】図５に公開鍵証明書発行局（ＩＡ）３０１が管理するデータの主要項目を示す。ＲＡＩＤは、サービス対象の登録局（ＲＡ）の識別子である。ＩＤは、認証対象者の識別子である。公開鍵は認証対象者の公開鍵、証明書は公開鍵証明書本体である。有効フラグは公開鍵証明書が有効であるか否かを示すフラグである。

【００６６】図６および図７に公開鍵証明書のフォーマット例を示す。これは、公開鍵証明書フォーマットＸ．５０９　Ｖ３に準拠した例である。

【００６７】バージョン（version）は、証明書フォーマットのバージョンを示す。シリアルナンバ（Serial Number）は、公開鍵証明書発行局（ＩＡ）によって設定される公開鍵証明書のシリアルナンバである。署名アルゴリズム識別子、アルゴリズムパラメータ（Signature algorithm Identifier algorithm parameter）は、公開鍵証明書の署名アルゴリズムとそのパラメータを記録するフィールドである。なお、署名アルゴリズムとしては、楕円曲線暗号およびＲＳＡがあり、楕円曲線暗号が適用されている場合はパラメータおよび鍵長が記録され、ＲＳＡが適用されている場合には鍵長が記録される。発行者（issuer）は、公開鍵証明書の発行者、すなわち公開鍵証明書発行局（ＩＡ）の名称が識別可能な形式（Distinguished Name）で記録されるフィールドである。有効期限(validity)は、証明書の有効期限である開始日時、終了日時が記録される。サブジェクト(subject)は、ユーザである認証対象者の名前が記録される。具体的には例えばユーザ機器のＩＤや、サービス提供主体のＩＤ等である。サブジェクト公開鍵情報（subject Public Key Info algorithm subject Public key）は、ユーザの公開鍵情報としての鍵アルゴリズム、鍵情報そのものを格納するフィールドである。

【００６８】ここまでが、公開鍵証明書フォーマットＸ．５０９　Ｖ１に含まれるフィールドであり、以下は、公開鍵証明書フォーマットＸ．５０９　Ｖ３において追加されるフィールドである。

【００６９】証明局鍵識別子（authority Key Identifier－key Identifier、authority Cert Issuer、authority Cert Serial Number）は、公開鍵証明書発行局（ＩＡ）の鍵を識別する情報であり、鍵識別番号（８進数）、公開鍵証明書発行局（ＩＡ）の名称、認証番号を格納する。サブジェクト鍵識別子（subject key Identifier）は、複数の鍵を公開鍵証明書において証明する場合に各鍵を識別するための識別子を格納する。鍵使用目的（key usage）は、鍵の使用目的を指定するフィールドであり、（０）ディジタル署名用、（１）否認防止用、（２）鍵の暗号化用、（３）メッセージの暗号化用、（４）共通鍵配送用、（５）認証の署名確認用、（６）失効リストの署名確認用の各使用目的が設定される。秘密鍵有効期限(private Key Usage Period)は、ユーザの有する秘密鍵の有効期限を記録する。認証局ポリシー（certificate Policies）は、認証局、ここでは、公開鍵証明書発行局（ＩＡ）および登録局（ＲＡ）の証明書発行ポリシーを記録する。例えばＩＳＯ／ＩＥＣ９３８４－１に準拠したポリシーＩＤ、認証基準である。ポリシー・マッピング（policy Mapping）は、ＣＡ（公開鍵証明書発行局（ＩＡ））を認証する場合にのみ記録するフィールドであり、証明書発行を行なう公開鍵証明書発行局（ＩＡ）のポリシーと、被認証ポリシーのマッピングを規定する。サポート・アルゴリズム（supported Algorithms）は、ディレクトリ（Ｘ．５００）のアトリビュートを定義する。これは、コミュニケーションの相手がディレクトリ情報を利用する場合に、事前にそのアトリビュートを知らせるのに用いる。サブジェクト別名（subject Alt Name）は、ユーザの別名を記録するフ

ィールドである。発行者別名（issuer Alt Name）は、証明書発行者の別名を記録するフィールドである。サブジェクト・ディレクトリ・アトリビュート（subject Directory Attribute）は、ユーザの任意の属性を記録するフィールドである。基本制約（basic Constraint）は、証明対象の公開鍵が認証局（公開鍵証明書発行局（ＩＡ））の署名用か、ユーザのものかを区別するためのフィールドである。許容サブトリー制約名（name Constraints permitted Subtrees）は、被認証者が認証局（公開鍵証明書発行局（ＩＡ））である場合にのみ使用される証明書の有効領域を示すフィールドである。制約ポリシー（policy Constraints）は、認証パスの残りに対する明確な認証ポリシーＩＤ、禁止ポリシーマップを要求する制限を記述する。ＣＲＬ参照ポイント（Certificate Revocation List Distribution Points）は、ユーザが証明書を利用する際に、証明書が失効していないか、どうかを確認するための失効リスト（図９参照）の参照ポイントを記述するフィールドである。署名は、公開鍵証明書発行者（公開鍵証明書発行局（ＩＡ））の署名フィールドである。

【００７０】図８に、図３、図４における登録局が有するエンティテイ・データベースのデータ構成を示す。エンティテイ・データベースは認証対象者の管理をするデータベースとして構成される。

【００７１】ＩＤは、認証対象者の識別子を格納するフィールドである。認証データは、認証対象者の認証に必要な情報、例えばユーザ端末が認証対象であれば、ユーザ端末ＩＤ等が記録される。認証結果は、最新の認証結果が記録される。失効情報は、失効リストへのポインタ情報が記録される。

【００７２】図９および図１０に失効リストのフォーマット例（Ｘ．５０９　Ｖ２準拠）を示す。図９は共通項目であり、図１０は個別の証明書毎に管理される情報である。各項目について説明する。

【００７３】署名アルゴリズム識別子（Signature Algorithm Identifier）は、適用される署名についての署名アルゴリズムを記述する。例えば楕円曲線暗号であるか、ＲＳＡであるか等である。発行者(Issuer)は、失効リストの発行者を記録する。図３、４の例では登録局名が記録される。更新日時(This Update)は、失効リストの発行日時（最新の更新日時）が記録される。次更新日時（Next Update）は、次の失効リスト更新日時を記録する。バージョン（Version）は、失効リストのバージョンを記録する。証明局鍵識別子（authority Key Identifier－key Identifier、authority Cert Issuer、authority Cert Serial Number）は、公開鍵証明書発行局（ＩＡ）の鍵を識別する情報であり、鍵識別番号（８進数）、公開鍵証明書発行局（ＩＡ）の名称、認証番号を格納する。ＣＲＬナンバ（CRL Number）は、失効リストの発行通し番号が記録される。失効リスト情報（Issuing distribution point）は、失効リストの各種情報を記録し、配布局名（Distribution point）、加入者の失効専用であるか否か（only contains user certs）、認証局（ＣＡ）（本例においては公開鍵証明書発行局（ＩＡ））の認証の失効専用であるか否か（only contains ＣＡ certs）、何らかのその他の失効理由に関するか否か（only some reasons）の情報、失効リストが間接失効リスト（indirect CRL）であるか否かが記録される。間接失効リスト（indirect CRL）とは、失効理由の情報管理と失効リスト管理が別々な機関で行われている態様である。例えばルートＲＡが失効リストを発行し、公開鍵証明書発行局（ＩＡ）が情報を管理している場合には、間接失効リスト（indirectCRL）として定義され、この場合、失効情報の格納ポイント、例えばＩＡの識別子を示すデータが格納される。本発明の構成では、失効リストは、間接失効リスト（indirect CRL）として生成され、失効理由の情報は、失効リスト発行局ではなく、認証発行局、すなわち公開鍵証明書発行局（ＩＡ）が管理する。ＣＲＬ識別子差分（Delta CRL Indicator）は、失効リストが差分リストとして配布される構成か否かのデータを記録する。差分リストとは、失効候補の公開鍵情報から失効確定の公開鍵情報を抜き出して関連プロバイダに提供可能としたリスト構成である。

【００７４】図１０は個別の証明書毎に管理される情報を示した図である。認証番号（certificate Serial Number）は、認証番号を記録する。リボケーション・デート（Revocation Date）は、失効申請受理日時を記録する。

【００７５】ここまでがバージョン１に規定されるフィールドであり、以下はバージョン２において規定されるフィールドである。

【００７６】理由コード（Reason code）は、失効理由を記述するフィールドである。失効理由としては、０：理由不明、１：加入者の鍵が危瀕を受けた、２：ＣＡ（公開鍵証明書発行局（ＩＡ））の鍵が危瀕を受けた、３：認証の情報が変更、４：当該認証が置き換えられた、５：利用中止、６：利用の一時中止、７：一時中止の状態解除が失効理由として規定される。対処方法（Hold instruction code）は、一時利用中止に対する対処方法を記述する。失効日（Invalidity date）は、秘密鍵が被害にあったと考えられる日時を記述する。認証発行局名（certificate issuer）は、認証発行局名を記述する。しかし、間接(indirect)失効リストの場合は、失効情報が失効リスト発行局で管理されていないため、指定された失効情報管理ＣＡ（例えば公開鍵証明書発行局（ＩＡ））に迂回、すなわちＩＡに対するポインタを設定する。署名は、失効リスト発行者の署名である。

【００７７】本発明の公開鍵証明書発行システムおよびデータ通信方法における処理において使用される電子署名および相互認証処理について説明する。電子署名およ

び相互認証処理について説明した後に、本発明の公開鍵証明書発行システムにおける具体的処理の詳細を、それぞれ図を用いて説明する。

【００７８】［電子署名］公開鍵暗号方式を用いた電子署名の生成方法を図１１を用いて説明する。図１１に示す処理は、ＥＣ－ＤＳＡ（(Elliptic Curve Digital Signature Algorithm)、IEEE P1363/D3）を用いた電子署名データの生成処理フローである。なお、ここでは公開鍵暗号として楕円曲線暗号（Elliptic Curve Cryptography（以下、ＥＣＣと呼ぶ））を用いた例を説明する。なお、本発明のデータ処理装置においては、楕円曲線暗号以外にも、同様の公開鍵暗号方式における、例えばＲＳＡ暗号（(Rivest、Shamir、Adleman)など（ANSI X9.31））を用いることも可能である。

【００７９】図１１の各ステップについて説明する。ステップＳ１において、ｐを標数、a、bを楕円曲線の係数（楕円曲線：$y^2=x^3+ax+b$）、Ｇを楕円曲線上のベースポイント、rをＧの位数、Ｋｓを秘密鍵（０＜Ｋｓ＜ｒ）とする。ステップＳ２おいて、メッセージＭのハッシュ値を計算し、f=Hash(M)とする。

【００８０】ここで、ハッシュ関数を用いてハッシュ値を求める方法を説明する。ハッシュ関数とは、メッセージを入力とし、これを所定のビット長のデータに圧縮し、ハッシュ値として出力する関数である。ハッシュ関数は、ハッシュ値（出力）から入力を予測することが難しく、ハッシュ関数に入力されたデータの1ビットが変化したとき、ハッシュ値の多くのビットが変化し、また、同一のハッシュ値を持つ異なる入力データを探し出すことが困難である特徴を有する。ハッシュ関数としては、ＭＤ４、ＭＤ５、ＳＨＡ－１などが用いられる場合もあるし、ＤＥＳ－ＣＢＣが用いられる場合もある。この場合は、最終出力値となるＭＡＣ（チェック値：ＩＣＶに相当する）がハッシュ値となる。

【００８１】続けて、ステップＳ３で、乱数ｕ（０＜ｕ＜ｒ）を生成し、ステップＳ４でベースポイントをｕ倍した座標Ｖ（Ｘｖ，Ｙｖ）を計算する。なお、楕円曲線上の加算、２倍算は次のように定義されている。

【００８２】

【数１】P=(Xa, Ya), Q=(Xb, Yb), R=(Xc, Yc)=P+Qとすると、P≠Qの時（加算）、

$Xc=\lambda^2-Xa-Xb$

$Yc=\lambda\times(Xa-Xc)-Ya$

$\lambda=(Yb-Ya)/(Xb-Xa)$

P=Qの時（2倍算）、

$Xc=\lambda^2-2Xa$

$Yc=\lambda\times(Xa-Xc)-Ya$

$\lambda=(3(Xa)^2+a)/(2Ya)$

【００８３】これらを用いて点Ｇのｕ倍を計算する（速度は遅いが、最もわかりやすい演算方法として次のように行う。Ｇ、２×Ｇ、４×Ｇ・・を計算し、uを2進数展開して1が立っているところに対応する$2^i\times G$（Ｇをｉ回２倍算した値（ｉはｕのＬＳＢから数えた時のビット位置））を加算する。

【００８４】ステップＳ５で、ｃ＝Ｘｖｍｏｄ　ｒを計算し、ステップＳ６でこの値が０になるかどうか判定し、０でなければステップＳ７でｄ＝［（ｆ＋ｃＫｓ）／ｕ］ｍｏｄ　ｒを計算し、ステップＳ８でｄが０であるかどうか判定し、ｄが０でなければ、ステップＳ９でｃおよびｄを電子署名データとして出力する。仮に、ｒを１６０ビット長の長さであると仮定すると、電子署名データは３２０ビット長となる。

【００８５】ステップＳ６において、ｃが０であった場合、ステップＳ３に戻って新たな乱数を生成し直す。同様に、ステップＳ８でｄが０であった場合も、ステップＳ３に戻って乱数を生成し直す。

【００８６】次に、公開鍵暗号方式を用いた電子署名の検証方法を、図１２を用いて説明する。ステップＳ１１で、Ｍをメッセージ、ｐを標数、a、bを楕円曲線の係数（楕円曲線：$y^2=x^3+ax+b$）、Ｇを楕円曲線上のベースポイント、rをＧの位数、ＧおよびＫｓ×Ｇを公開鍵（０＜Ｋｓ＜ｒ）とする。ステップＳ１２で電子署名データｃおよびｄが０＜ｃ＜ｒ、０＜ｄ＜ｒを満たすか検証する。これを満たしていた場合、ステップＳ１３で、メッセージＭのハッシュ値を計算し、ｆ＝Ｈａｓｈ（Ｍ）とする。次に、ステップＳ１４でh=1/d mod rを計算し、ステップＳ１５でｈ１＝ｆｈ　ｍｏｄ　ｒ、ｈ２＝ｃｈ　ｍｏｄ　ｒを計算する。

【００８７】ステップＳ１６において、既に計算したｈ１およびｈ２を用い、点Ｐ＝（Ｘｐ，Ｙｐ）＝ｈ１×Ｇ＋ｈ２・Ｋｓ×Ｇを計算する。電子署名検証者は、公開鍵ＧおよびＫｓ×Ｇを知っているので、図１１のステップＳ４と同様に楕円曲線上の点のスカラー倍の計算ができる。そして、ステップＳ１７で点Ｐが無限遠点かどうか判定し、無限遠点でなければステップＳ１８に進む（実際には、無限遠点の判定はステップＳ１６でできてしまう。つまり、Ｐ＝（Ｘ，Ｙ）、Ｑ＝（Ｘ，－Ｙ）の加算を行うと、λが計算できず、Ｐ＋Ｑが無限遠点であることが判明している）。ステップＳ１８でＸｐ　ｍｏｄ　ｒを計算し、電子署名データｃと比較する。最後に、この値が一致していた場合、ステップＳ１９に進み、電子署名が正しいと判定する。

【００８８】電子署名が正しいと判定された場合、データは改竄されておらず、公開鍵に対応した秘密鍵を保持する者が電子署名を生成したことがわかる。

【００８９】ステップＳ１２において、電子署名データｃまたはｄが、０＜ｃ＜ｒ、０＜ｄ＜ｒを満たさなかった場合、ステップＳ２０に進む。また、ステップＳ１７において、点Ｐが無限遠点であった場合もステップＳ２０に進む。さらにまた、ステップＳ１８において、Ｘｐ　ｍｏｄ　ｒの値が、電子署名データｃと一致していなか

った場合にもステップＳ２０に進む。

【００９０】ステップＳ２０において、電子署名が正しくないと判定された場合、データは改竄されているか、公開鍵に対応した秘密鍵を保持する者が電子署名を生成したのではないことがわかる。

【００９１】［相互認証処理］データ送受信を実行する２つの手段間では、相互に相手が正しいデータ通信者であるか否かを確認して、その後に必要なデータ転送を行なうことが行われる。相手が正しいデータ通信者であるか否かの確認処理が相互認証処理である。相互認証処理時にセッション鍵の生成を実行して、生成したセッション鍵を共有鍵として暗号化処理を実行してデータ送信を行なう構成が１つの好ましいデータ転送方式である。

【００９２】共通鍵暗号方式を用いた相互認証方法を、図１３を用いて説明する。図１３において、共通鍵暗号方式としてＤＥＳを用いているが、同様な共通鍵暗号方式であればいずれでもよい。

【００９３】まず、Ｂが６４ビットの乱数Ｒｂを生成し、Ｒｂおよび自己のＩＤであるＩＤ（ｂ）をＡに送信する。これを受信したＡは、新たに６４ビットの乱数Ｒａを生成し、Ｒａ、Ｒｂ、ＩＤ（ｂ）の順に、ＤＥＳのＣＢＣモードで鍵Ｋａｂを用いてデータを暗号化し、Ｂに返送する。

【００９４】これを受信したＢは、受信データを鍵Ｋａｂで復号化する。受信データの復号化方法は、まず、暗号文Ｅ１を鍵Ｋａｂで復号化し、乱数Ｒａを得る。次に、暗号文Ｅ２を鍵Ｋａｂで復号化し、その結果とＥ１を排他的論理和し、Ｒｂを得る。最後に、暗号文Ｅ３を鍵Ｋａｂで復号化し、その結果とＥ２を排他的論理和し、ＩＤ(b)を得る。こうして得られたＲａ、Ｒｂ、ＩＤ(b)の内、ＲｂおよびＩＤ(b)が、Ｂが送信したものと一致するか検証する。この検証に通った場合、ＢはＡを正当なものとして認証する。

【００９５】次にＢは、認証後に使用するセッション鍵（Session Key（以下、Ｋｓｅｓとする））を生成する（生成方法は、乱数を用いる）。そして、Ｒｂ、Ｒａ、Ｋｓｅｓの順に、ＤＥＳのＣＢＣモードで鍵Ｋａｂを用いて暗号化し、Ａに返送する。

【００９６】これを受信したＡは、受信データを鍵Ｋａｂで復号化する。受信データの復号化方法は、Ｂの復号化処理と同様であるので、ここでは詳細を省略する。こうして得られたＲｂ、Ｒａ、Ｋｓｅｓの内、ＲｂおよびＲａが、Ａが送信したものと一致するか検証する。この検証に通った場合、ＡはＢを正当なものとして認証する。互いに相手を認証した後には、セッション鍵Ｋｓｅｓは、認証後の秘密通信のための共通鍵として利用される。

【００９７】なお、受信データの検証の際に、不正、不一致が見つかった場合には、相互認証が失敗したものとして処理を中断する。

【００９８】次に、公開鍵暗号方式である１６０ビット長の楕円曲線暗号を用いた相互認証方法を、図１４を用いて説明する。図１４において、公開鍵暗号方式としてＥＣＣを用いているが、前述のように同様な公開鍵暗号方式であればいずれでもよい。また、鍵サイズも１６０ビットでなくてもよい。図１４において、まずＢが、６４ビットの乱数Ｒｂを生成し、Ａに送信する。これを受信したＡは、新たに６４ビットの乱数Ｒａおよび標数ｐより小さい乱数Ａｋを生成する。そして、ベースポイントＧをＡｋ倍した点Ａｖ＝Ａｋ×Ｇを求め、Ｒａ、Ｒｂ、Ａｖ（X座標とY座標）に対する電子署名Ａ．Ｓｉｇを生成し、Ａの公開鍵証明書とともにＢに返送する。ここで、ＲａおよびＲｂはそれぞれ６４ビット、ＡｖのＸ座標とＹ座標がそれぞれ１６０ビットであるので、合計４４８ビットに対する電子署名を生成する。電子署名の生成方法は図１１で説明したので、その詳細は省略する。

【００９９】公開鍵証明書を利用する際には、利用者は自己が保持する公開鍵証明書発行局（ＩＡ）４１０の公開鍵を用い、当該公開鍵証明書の電子署名を検証し、電子署名の検証に成功した後に公開鍵証明書から公開鍵を取り出し、当該公開鍵を利用する。従って、公開鍵証明書を利用する全ての利用者は、共通の公開鍵証明書発行局（ＩＡ）の公開鍵を保持している必要がある。なお、電子署名の検証方法については、図１２で説明したのでその詳細は省略する。

【０１００】図１４に戻って説明を続ける。Ａの公開鍵証明書、Ｒａ、Ｒｂ、Ａｖ、電子署名Ａ．Ｓｉｇを受信したＢは、Ａが送信してきたＲｂが、Ｂが生成したものと一致するか検証する。その結果、一致していた場合には、Ａの公開鍵証明書内の電子署名を認証局の公開鍵で検証し、Ａの公開鍵を取り出す。そして、取り出したＡの公開鍵を用い電子署名Ａ．Ｓｉｇを検証する。電子署名の検証方法は図１２で説明したので、その詳細は省略する。電子署名の検証に成功した後、ＢはＡを正当なものとして認証する。

【０１０１】次に、Ｂは、標数ｐより小さい乱数Ｂｋを生成する。そして、ベースポイントＧをＢｋ倍した点Ｂｖ＝Ｂｋ×Ｇを求め、Ｒｂ、Ｒａ、Ｂｖ（X座標とY座標）に対する電子署名Ｂ．Ｓｉｇを生成し、Ｂの公開鍵証明書とともにＡに返送する。

【０１０２】Ｂの公開鍵証明書、Ｒｂ、Ｒａ、Ａｖ、電子署名Ｂ．Ｓｉｇを受信したＡは、Ｂが送信してきたＲａが、Ａが生成したものと一致するか検証する。その結果、一致していた場合には、Ｂの公開鍵証明書内の電子署名を認証局の公開鍵で検証し、Ｂの公開鍵を取り出す。そして、取り出したＢの公開鍵を用い電子署名Ｂ．Ｓｉｇを検証する。電子署名の検証に成功した後、ＡはＢを正当なものとして認証する。

【０１０３】両者が認証に成功した場合には、ＢはＢｋ

×Ａｖ（Ｂｋは乱数だが、Ａｖは楕円曲線上の点であるため、楕円曲線上の点のスカラー倍計算が必要）を計算し、ＡはＡｋ×Ｂｖを計算し、これら点のＸ座標の下位６４ビットをセッション鍵として以降の通信に使用する（共通鍵暗号を６４ビット鍵長の共通鍵暗号とした場合）。もちろん、Ｙ座標からセッション鍵を生成してもよいし、下位６４ビットでなくてもよい。なお、相互認証後の秘密通信においては、送信データはセッション鍵で暗号化されるだけでなく、電子署名も付されることがある。

【０１０４】電子署名の検証や受信データの検証の際に、不正、不一致が見つかった場合には、相互認証が失敗したものとして処理を中断する。

【０１０５】このような相互認証処理において、生成したセッション鍵を用いて、送信データを暗号化して、相互にデータ通信を実行する。

【０１０６】図１５に、本発明の公開鍵証明書発行システムおよびデータ通信方法に関する以下の説明において使用される用語についての説明を示す。これらについて簡単に説明する。鍵をＫとして表記し、サフィックスとして公開鍵はＰ、秘密鍵はＳを付加し、さらに所有者識別子（例えばａ）を付加する。相互認証の際に生成され、暗号化、復号化処理に適用されるセッション鍵をＫｓとする。Ａが発行したＢの公開鍵証明書をＡ≪Ｂ≫とする。データの暗号化は、例えばセッション鍵Ｋｓでデータ（ｄａｔａ）を暗号化した場合は、E_{Ks}（ｄａｔａ）として示す。同様の復号は、D_{Ks}（ｄａｔａ）として示す。署名処理は、例えばデータ（ｄａｔａ）をＡの秘密鍵Ｋｓａで署名した場合は、｛ｄａｔａ｝Ｓｉｇ・Ｋｓａとして示す。また、署名付き暗号化データは、例えばデータ（ｄａｔａ）をＡの秘密鍵Ｋｓａで署名して生成される（ｄａｔａ‖署名）をセッション鍵Ｋｓで暗号化した場合は、E_{Ks}（｛ｄａｔａ｝Ｓｉｇ・Ｋｓａ）で示す。

【０１０７】図１６に、ルート登録局（ルートＲＡ）の登録処理の際に実行される処理を説明した図を示す。（ａ）はオンラインで行なう場合、（ｂ）はオフラインで行なう場合である。処理順は番号で示してある。

（ａ）のオンライン処理について説明する。オンライン処理の場合、図には示されていないが、先に別ルートで事前にルート登録局（ルートＲＡ）に渡されているルート登録局（ルートＲＡ）の公開鍵ペアを用いて図１３、図１４で説明した相互認証がルート登録局（ルートＲＡ）１６０１と公開鍵証明書発行局（ＩＡ）１６０２間で実行され、通信相手の確認が行われ、セッション鍵Ｋｓが生成される。

【０１０８】相互認証処理が終了すると、ルート登録局（ルートＲＡ）１６０１が自己の公開鍵、秘密鍵を生成して、生成した公開鍵に関する公開鍵証明書の発行要求を公開鍵証明書発行局（ＩＡ）１６０２に対して行なう。証明書の発行要求は、ルート登録局（ルートＲＡ）１６０１の識別子（ＲｏｏｔＲＡＩＤ）と、ルート登録局（ルートＲＡ）１６０１の公開鍵（ＫｐＲｏｏｔＲＡ’）に対してセッション鍵Ｋｓで暗号化したデータE_{Ks}（｛ＲｏｏｔＲＡＩＤ，ＫｐＲｏｏｔＲＡ’｝$_{Sig\text{-}KsRootRA}$）をルート登録局（ルートＲＡ）１６０１から公開鍵証明書発行局（ＩＡ）１６０２に対して転送することによって行われる。

【０１０９】公開鍵証明書発行局（ＩＡ）１６０２は、受信した証明書発行要求である暗号化データE_{Ks}（｛ＲｏｏｔＲＡＩＤ，ＫｐＲｏｏｔＲＡ’｝$_{Sig\text{-}KsRootRA}$）を復号して署名を健勝した後、ルート登録局（ルートＲＡ）１６０１の公開鍵ＫｐＲＡを管理データとしてデータベースに格納する。すなわち、先に図５を用いて説明したデータベース中のデータとして格納する。

【０１１０】公開鍵証明書発行局（ＩＡ）１６０２は、さらに、ルート登録局（ルートＲＡ）１６０１からの証明書発行要求に従って、公開鍵証明書を作成する。これは、先の図６、図７の態様に従った公開鍵証明書である。公開鍵証明書発行局（ＩＡ）１６０２は、生成した公開鍵証明書をデータベースに格納して管理（図５参照）するとともに、公開鍵証明書発行局（ＩＡ）１６０２の発行したルート登録局（ルートＲＡ）１６０１の公開鍵証明書、すなわちＩＡ≪ＲｏｏｔＲＡ≫に対して、公開鍵証明書発行局（ＩＡ）１６０２の秘密鍵ＫｓＩＡを用いて署名を行ない、先の相互認証処理において生成したセッション鍵Ｋｓで暗号化を行なって、表記データE_{Ks}（｛ＩＡ≪ＲｏｏｔＲＡ≫｝$_{Sig\text{-}KsIA}$）を生成して、ルート登録局（ルートＲＡ）１６０１に送信する。ルート登録局（ルートＲＡ）１６０１は、これを格納する。事前に共有する秘密情報は共通鍵でもよい。その場合、ＳｉｇＫｓＲｏｏｔＲＡは、共通鍵を用いたＭＡＣ値となる。

【０１１１】なお、公開鍵証明書発行局（ＩＡ）１６０２がルート登録局を介さずにダイレクトに登録局（ＲＡ）との間で公開鍵証明書を発行する場合は、図１６のルート登録局（ルートＲＡ）１６０１を登録局（ＲＡ）に置き換えた処理となる。

【０１１２】図１６（ｂ）はオフライン、例えばＤＶＤ、ＣＤ、メモリカード等の各種の記憶媒体を介して登録処理を実行する場合であり、オフライン処理の場合は、記憶媒体中に上記オンライン処理で説明した情報を格納して処理を実行する。

【０１１３】図１７は、ルート登録局（ルートＲＡ）１６０１の管理下に例えば音楽データや画像データ等のコンテンツ配信サービスを行なうサービスプロバイダ、あるいは電子マネーの決算処理を行なうクリアリングセンタとしてのサービスプロバイダとしての登録局（ＲＡ）１７０１がある場合、登録局（ＲＡ）１７０１が公開鍵

の証明書発行を行なう場合の手続き例を示したものである。手続き順に従って説明する。

【０１１４】まず、登録局（ＲＡ）１７０１と、ルート登録局（ルートＲＡ）１６０１の間で、相互認証処理が実行される。これは、例えば登録局（ＲＡ）１７０１と、ルート登録局（ルートＲＡ）１６０１の双方のメモリに予め格納されている相互認証用鍵（図１３で説明した鍵Ｋａｂに相当）を用いて行なう。

【０１１５】次に、登録局（ＲＡ）１７０１は、登録局（ＲＡ）１７０.1の識別子ＳＰＩＤに自己の秘密鍵K_{SSP}を用いて署名を行ない、さらに、相互認証時に生成したセッション鍵Ｋｓを用いて暗号化を行なってデータE_{Ks}（$\{SPID\}_{Sig\cdot KsSP}$）を生成してルート登録局（ルートＲＡ）１６０１に送信する。ルート登録局（ルートＲＡ）１６０１は、受信データをセッション鍵Ｋｓで復号して署名を検証後、識別子ＳＰＩＤを確認して確認結果に署名処理を実行して、さらにセッション鍵で暗号化して登録局（ＲＡ）１７０１に対する確認応答を実行する。

【０１１６】ルート登録局（ルートＲＡ）１６０１からの確認応答を受け取った登録局（ＲＡ）１７０１は、自己の公開鍵、秘密鍵を生成して、公開鍵ＫｐＳＰ’を自己の秘密鍵で署名して、さらにセッション鍵で暗号化してデータE_{Ks}（$\{KpSp'\}_{Sig\cdot KsSP}$）を生成してルート登録局（ルートＲＡ）１６０１に送信する。このデータを受領したルート登録局（ルートＲＡ）１６０１は、公開鍵証明書発行局（ＩＡ）１６０２に対して登録局（ＲＡ）１７０１の証明書発行要求を実行する。さらに、ルート登録局（ルートＲＡ）１６０１と公開鍵証明書発行局（ＩＡ）１６０２との間で相互認証処理が実行される。

【０１１７】相互認証処理において、認証が成立すると、ルート登録局（ルートＲＡ）１６０１は、登録局（ＲＡ）１７０１の識別子、ＳＰＩＤと登録局（ＲＡ）１７０１の公開鍵ＫｐＳｐにルート登録局（ルートＲＡ）１６０１の秘密鍵で署名を実行して相互認証時に生成したセッション鍵で暗号化してE_{Ks}（$\{SPID, KpSP'\}_{Sig\cdot KsRootSP}$）を生成して公開鍵証明書発行局（ＩＡ）１６０２に送信する。

【０１１８】公開鍵証明書発行局（ＩＡ）１６０２は、受信データE_{Ks}（$\{SPID, KpSP'\}_{Sig\cdot KsRootSP}$）を復号して、登録局（ＲＡ）１７０１の公開鍵ＫｐＳＰを管理データとしてデータベースに格納する。すなわち、先に図５を用いて説明したデータベース中のデータとして格納する。

【０１１９】公開鍵証明書発行局（ＩＡ）１６０２は、さらに、登録局（ＲＡ）１７０１の公開鍵証明書を作成する。これは、先の図６、図７の態様に従った公開鍵証明書である。公開鍵証明書発行局（ＩＡ）１６０２は、生成した公開鍵証明書をデータベースに格納して管理（図５参照）するとともに、公開鍵証明書発行局（ＩＡ）１６０２の発行した登録局（ＲＡ）１７０１の公開鍵証明書、すなわちＩＡ≪ＳＰ≫に対して、公開鍵証明書発行局（ＩＡ）１６０２の秘密鍵ＫｓＩＡを用いて署名を行ない、先の相互認証処理において生成したセッション鍵Ｋｓ２で暗号化を行なって、データE_{Ks}（$\{IA\ll SP\gg\}_{Sig\cdot KsIA}$）を生成して、ルート登録局（ルートＲＡ）１６０１に送信する。ルート登録局（ルートＲＡ）１６０１は、署名を検討して、さらに、自己の秘密鍵による署名を行ない、これをセッション鍵（登録局（ＲＡ）１７０１とルート登録局（ルートＲＡ）１６０１の間で実行された相互認証処理において生成されたセッション鍵）で暗号化して登録局（ＲＡ）１７０１に送信する。登録局（ＲＡ）１７０１は、受信データのセッション鍵での復号、署名検証処理後、証明書を格納する。

【０１２０】なお、上述の手続き中⑤の鍵生成処理を省略して、先に登録局（ＲＡ）１７０１のデバイス中に埋め込まれている鍵をそのまま公開鍵として適用する構成としてもよい。また、初期埋め込み鍵は共通鍵とし、相互認証を行ない、②への署名はその共通鍵を用いて生成したＭＡＣとしてもよい。

【０１２１】図１８は、図１７で説明した登録局（ＲＡ）１７０１の公開鍵証明書の発行処理をオフラインにおいて実行する構成を説明しているものであり、図１７の各局間で送信されるデータを、例えばＤＶＤ、ＣＤ、メモリカード等の各種の記憶媒体を介して受け渡しを行なって処理する。オフライン処理の場合は、記憶媒体中に上記オンライン処理で説明した情報を格納して処理を実行する。

【０１２２】図１９は、登録局（ＲＡ）１７０１の管理下に、例えばコンテンツの利用を行なうユーザ、コンテンツ販売を行なうショップ等がある場合、ユーザ（ショップ等含む）２００１が公開鍵の証明書発行を行なう場合の手続き例を示したものである。手続き順に従って説明する。

【０１２３】ユーザ２００１のデバイスには、初期埋め込み鍵として、ユーザデバイスの公開鍵ＫｐＵＤ、秘密鍵ＫｓＵＤ、さらに登録局（ＲＡ）１７０１の公開鍵ＫｐＲＡ、公開鍵証明書発行局（ＩＡ）１６０２の公開鍵ＫｐＩＡが例えばＳＡＭ（secure Application module）内のメモリに埋め込まれている。

【０１２４】まず、ユーザ２００１と登録局（ＲＡ）１７０１との間で相互認証処理が実行され、相互認証処理においてセッション鍵が生成される。これは、例えばユーザ２００１に予め格納されているＫｐＵＤを用いて行なう。

【０１２５】次に、ユーザ２００１は、ユーザ２００１の識別子ＳＡＭＩＤに自己の秘密鍵K_{SUD}を用いて署名を行ない、さらに、相互認証時に生成したセッション鍵

Ｋｓを用いて暗号化を行なってデータE_{Ks}（{ＳＰＩＤ}$_{Sig-KsUD}$）を生成してルート登録局（ＲＡ）１７０１に送信する。登録局（ＲＡ）１７０１は、受信データをセッション鍵Ｋｓで復号して、識別子ＳＡＭＩＤを確認して確認結果に署名処理を実行して、さらにセッション鍵で暗号化してユーザ２００１に対する確認応答を実行する。

【０１２６】登録局（ＲＡ）１７０１からの確認応答を受け取ったユーザ２００１は、自己の公開鍵、秘密鍵を生成して、公開鍵ＫｐＵＤ’を自己の秘密鍵ＫｓＵＤで署名して、さらにセッション鍵で暗号化してデータE_{Ks}（{ＫｐＵＤ’}$_{Sig-KsUD}$）を生成して登録局（ＲＡ）１７０１に送信する。

【０１２７】データを受領した登録局（ＲＡ）１７０１は、ルート登録局（ルートＲＡ）１６０１との間で相互認証を実行してセッション鍵生成を行なう。次に、登録局（ＲＡ）１７０１は、ユーザ２００１の識別子ＳＡＭＩＤと、公開鍵ＫｐＵＤとに自己の秘密鍵ＫｓＲＡを用いて署名を行ない、さらにセッション鍵Ｋｓ２で暗号化してルート登録局（ルートＲＡ）１６０１に送信する。

【０１２８】ルート登録局（ルートＲＡ）１６０１は、登録局（ＲＡ）１７０１から送信されてきたデータに関するセッション鍵での復号、署名検証処理を行ない、さらに、公開鍵証明書発行局（ＩＡ）１６０２に対してユーザ２００１の証明書発行要求を実行する。さらに、ルート登録局（ルートＲＡ）１６０１と公開鍵証明書発行局（ＩＡ）１６０２との間で相互認証処理が実行される。

【０１２９】相互認証処理において、認証が成立すると、ルート登録局（ルートＲＡ）１６０１は、ユーザ２００１の識別子、ＳＡＭＩＤと公開鍵ＫｐＵＤにルート登録局（ルートＲＡ）１６０１の秘密鍵で署名を実行して相互認証時に生成したセッション鍵で暗号化してE_{Ks3}（{ＳＡＭＩＤ，ＫｐＵＤ}$_{Sig-KsRootSP}$）を生成して公開鍵証明書発行局（ＩＡ）１６０２に送信する。

【０１３０】公開鍵証明書発行局（ＩＡ）１６０２は、受信データE_{Ks2}（{ＳＡＭＩＤ，ＫｐＵＤ}$_{Sig-KsRootSP}$）を復号して署名を検証後、ユーザ２００１の公開鍵ＫｐＵＤを管理データとしてデータベースに格納する。すなわち、先に図５を用いて説明したデータベース中のデータとして格納する。

【０１３１】公開鍵証明書発行局（ＩＡ）１６０２は、さらに、ユーザ２００１の公開鍵証明書を作成する。これは、先の図６、図７の態様に従った公開鍵証明書である。公開鍵証明書発行局（ＩＡ）１６０２は、生成した公開鍵証明書をデータベースに格納して管理（図５参照）するとともに、公開鍵証明書発行局（ＩＡ）１６０２の発行したユーザ２００１の公開鍵証明書、すなわちＩＡ≪ＵＤ≫に対して、公開鍵証明書発行局（ＩＡ）１６０２の秘密鍵ＫｓＩＡを用いて署名を行ない、先の相互認証処理において生成したセッション鍵Ｋｓ３で暗号化を行なって、データE_{Ks3}（{ＩＡ≪ＵＤ≫}$_{Sig-KsIA}$）を生成して、ルート登録局（ルートＲＡ）１６０１に送信する。ルート登録局（ルートＲＡ）１６０１は、署名を検証して、さらに、自己の秘密鍵による署名を行ない、これをセッション鍵（登録局（ＲＡ）１７０１とルート登録局（ルートＲＡ）１６０１の間で実行された相互認証処理において生成されたセッション鍵）で暗号化して登録局（ＲＡ）１７０１に送信する。登録局（ＲＡ）１７０１は、さらに、自己の秘密鍵による署名を行ない、これをセッション鍵（ユーザ２００１と登録局（ＲＡ）１７０１との間で実行された相互認証処理において生成されたセッション鍵）で暗号化してユーザ２００１に送信する。ユーザ２００１は、受信データのセッション鍵での復号、署名検証処理後、証明書を格納する。

【０１３２】なお、上述の手続き中⑤の鍵生成処理を省略して、先にユーザ２００１のデバイス中に埋め込まれている鍵をそのまま公開鍵として適用する構成としてもよい。また、初期埋め込み鍵は、共通鍵として相互認証を行ない、②への署名はその共通鍵を用いて生成したＭＡＣとしてもよい。

【０１３３】図２０は、ルート登録局（ルートＲＡ）１６０１の管理下の登録局（ＲＡ）１７０１であるサービスプロバイダ（ＳＰ）が新たな公開鍵、秘密鍵を生成して、新たに生成した公開鍵の証明書の発行を行なう手続き、すなわち更新処理を示した図である。

【０１３４】まず、登録局（ＲＡ）１７０１は、新たな公開鍵、秘密鍵を生成する。次に登録局（ＲＡ）１７０１とルート登録局（ルートＲＡ）１６０１との間で相互認証処理を実行する。これは、現鍵（現在（更新前）の公開鍵、秘密鍵）を用いた相互認証処理、すなわち図１４で説明した非対象鍵暗号を用いた処理として実行できる。

【０１３５】次に、登録局（ＲＡ）１７０１は、登録局（ＲＡ）１７０１が新たに生成した公開鍵ＫｐＳＰ’を自己の秘密鍵で署名して、さらに相互認証時に生成したセッション鍵で暗号化してデータE_{Ks}（{ＫｐＳｐ’}$_{Sig-KsSP}$）を生成してルート登録局（ルートＲＡ）１６０１に送信する。このデータを受領したルート登録局（ルートＲＡ）１６０１は、失効チェックを実行する。失効チェックは、先の図９、図１０で説明した失効リストに失効する公開鍵データを書き込む処理として実行される。

【０１３６】さらに、ルート登録局（ルートＲＡ）１６０１は、公開鍵証明書発行局（ＩＡ）１６０２に対して登録局（ＲＡ）１７０１の証明書発行要求を実行する。さらに、ルート登録局（ルートＲＡ）１６０１と公開鍵証明書発行局（ＩＡ）１６０２との間で相互認証処理が実行される。

【０１３７】相互認証処理において、認証が成立すると、ルート登録局（ルートＲＡ）１６０１は、登録局（ＲＡ）１７０１の識別子、ＳＰＩＤと登録局（ＲＡ）１７０１の公開鍵ＫｐＳｐにルート登録局（ルートＲＡ）１６０１の秘密鍵で署名を実行して相互認証時に生成したセッション鍵で暗号化してE_{Ks2}（{ＳＰＩＤ，ＫｐＳＰ’}$_{Sig-KsRootSP}$）を生成して公開鍵証明書発行局（ＩＡ）１６０２に送信する。

【０１３８】公開鍵証明書発行局（ＩＡ）１６０２は、受信データE_{Ks2}（{ＳＰＩＤ，ＫｐＳＰ’}$_{Sig-KsRootSP}$）を復号して署名検証を行なった後、登録局（ＲＡ）１７０１の公開鍵ＫｐＳＰの有効性チェックを行なう。有効性チェックは、先の図５で説明したデータベース中に同じユーザの公開鍵、公開鍵証明書が格納されている場合、これを無効化して、新たな更新された公開鍵を有効化する処理として実行される。公開鍵証明書発行局（ＩＡ）１６０２は、更新された新たな公開鍵の証明書を発行して、データベースに登録する。

【０１３９】公開鍵証明書発行局（ＩＡ）１６０２は、生成した公開鍵証明書、すなわちＩＡ≪ＳＰ≫に対して、公開鍵証明書発行局（ＩＡ）１６０２の秘密鍵ＫｓＩＡを用いて署名を行ない、先の相互認証処理において生成したセッション鍵Ｋｓで暗号化を行なって、データE_{Ks2}（{ＩＡ≪ＳＰ≫}$_{Sig-KsIA}$）を生成して、ルート登録局（ルートＲＡ）１６０１に送信する。ルート登録局（ルートＲＡ）１６０１は、署名を検証して、さらに、自己の秘密鍵による署名を行ない、これをセッション鍵（登録局（ＲＡ）１７０１とルート登録局（ルートＲＡ）１６０１の間で実行された相互認証処理において生成されたセッション鍵）で暗号化して登録局（ＲＡ）１７０１に送信する。登録局（ＲＡ）１７０１は、受信データのセッション鍵での復号、署名検証処理後、証明書を格納する。

【０１４０】図２１は、図２０と同様、ルート登録局（ルートＲＡ）１６０１の管理下の登録局（ＲＡ）１７０１であるサービスプロバイダ（ＳＰ）の新たな公開鍵、秘密鍵の証明書の発行を行なう手続きを示した図であるが、この図２１においては、サービスプロバイダ（ＳＰ）の新たな公開鍵、秘密鍵はルート登録局（ルートＲＡ）１６０１が生成する。

【０１４１】図２１において、手続き②～⑥が図２０と異なっている。これらの手続きについて説明する。登録局（ＲＡ）１７０１と、ルート登録局（ルートＲＡ）１６０１の間で、相互認証処理が実行される。これは、例えば登録局（ＲＡ）１７０１と、ルート登録局（ルートＲＡ）１６０１の双方のメモリに予め格納されている相互認証用鍵（図１３で説明した鍵Ｋａｂに相当）、あるいは現公開鍵、秘密鍵を用いて行なう（図１４参照）。

【０１４２】次に、登録局（ＲＡ）１７０１は、登録局（ＲＡ）１７０１の識別子ＳＰＩＤに自己の秘密鍵K_{SSP}を用いて署名を行ない、さらに、相互認証時に生成したセッション鍵Ｋｓを用いて暗号化を行なってデータE_{Ks}（{ＳＰＩＤ}$_{Sig-KsSP}$）を生成してルート登録局（ルートＲＡ）１６０１に送信する。ルート登録局（ルートＲＡ）１６０１は、受信データをセッション鍵Ｋｓで復号して署名を検証後、識別子ＳＰＩＤを確認する。確認がなされると、ルート登録局（ルートＲＡ）１６０１は、登録局（ＲＡ）１７０１の新たな公開鍵、秘密鍵を生成する。その後、ルート登録局（ルートＲＡ）１６０１は、生成した登録局（ＲＡ）１７０１の新たな公開鍵、秘密鍵（ＫｐＳＰ’，ＫｓＳＰ’）に自己の秘密鍵で署名処理を実行して、さらにセッション鍵で暗号化して登録局（ＲＡ）１７０１に対して送信する。その後の処理は、図２０と同様である。

【０１４３】次に、図２２を用いて公開鍵証明書の失効処理について説明する。図２２では、ユーザ２００１と、ルート登録局（ルートＲＡ）１６０１と、公開鍵証明書発行局（ＩＡ）１６０２との間の処理として示しているが、ユーザ２００１と、ルート登録局（ルートＲＡ）１６０１間に登録局１７０１がある場合は、登録局１７０１がユーザ２００１とルート登録局（ルートＲＡ）１６０１間の通信に介在することになる。

【０１４４】図２２の処理について説明する。ルート登録局（ルートＲＡ）１６０１は、ユーザ２００１の公開鍵が例えば不正に流通している、あるいはユーザ２００１からの申し出等により、失効させる処理を行なう。これは、先の図９、図１０で説明した失効リストにユーザ２００１の公開鍵情報を追加する処理として実行する。ルート登録局（ルートＲＡ）１６０１は失効リストに登録する処理を行なうとともに、公開鍵証明書発行局（ＩＡ）１６０２に公開鍵証明書の無効化を依頼する。

【０１４５】まず、ルート登録局（ルートＲＡ）１６０１と公開鍵証明書発行局（ＩＡ）１６０２との間で相互認証処理が実行される。相互認証が成立すると、ルート登録局（ルートＲＡ）１６０１は、失効した公開鍵のユーザ２００１の識別子ＳＡＭＩＳＤと、公開鍵ＫｐＵＤにルート登録局（ルートＲＡ）１６０１の秘密鍵で署名を実行して相互認証時に生成したセッション鍵で暗号化してE_{Ks}（{ＳＡＭＩＤ，ＫｐＵＤ}$_{Sig-KsRootSP}$）を生成して公開鍵証明書発行局（ＩＡ）１６０２に送信する。

【０１４６】公開鍵証明書発行局（ＩＡ）１６０２は、受信データE_{Ks}（{ＳＡＭＩＤ，ＫｐＵＤ}$_{Sig-KsRootSP}$）を復号して署名を検証後、ユーザ２００１の公開鍵ＫｐＵＤに対応する公開鍵証明書の無効化処理を行なう。すなわち、先に図５を用いて説明したデータベース中のフラグを無効に設定する。さらに、公開鍵証明書発行局（ＩＡ）１６０２は、無効化処理が実行されたか否か（ＯＫまたはＮＧ）の応答を署名およびセッション鍵による暗号化を行なったデータE_{Ks}（{ＯＫ

／ＮＧ｝$_{Sig-KsIA}$）をルート登録局（ルートＲＡ）１６０１に送信する。

【０１４７】これらの一連の公開鍵証明書の失効処理が終了すると、ユーザ２００１の公開鍵はルート登録局（ルートＲＡ）１６０１の管理するサービス下では使用できなくなる。すなわち、公開鍵を用いた暗号化データの送受信、認証処理等は、ルート登録局（ルートＲＡ）１６０１との間で成立せず、また、ルート登録局（ルートＲＡ）１６０１の管理する他のサービスプロバイダとの間でも、失効した公開鍵を用いた取り引きは成立しないことになる。ルート登録局（ルートＲＡ）は必要に応じて失効リストの差分配布を行なう。

【０１４８】図２３は、失効した公開鍵、公開鍵証明書の失効解除処理を説明するものである。ユーザ２００１は、公開鍵の失効状態においては、ルート登録局（ルートＲＡ）１６０１に対するアクセスが拒否（ＮＧ）される。ルート登録局（ルートＲＡ）１６０１は、失効した公開鍵の失効を解除する場合、公開鍵証明書発行局（ＩＡ）１６０２に証明書無効化要求を発行して、さらにルート登録局（ルートＲＡ）１６０１と公開鍵証明書発行局（ＩＡ）１６０２間での相互認証処理を実行する。

【０１４９】相互認証が成立すると、ルート登録局（ルートＲＡ）１６０１は、失効を解除する公開鍵のユーザ２００１の識別子ＳＡＭＩＳＤと、公開鍵ＫｐＵＤにルート登録局（ルートＲＡ）１６０１の秘密鍵で署名を実行して相互認証時に生成したセッション鍵で暗号化してE_{Ks}（｛ＳＡＭＩＤ，ＫｐＵＤ｝$_{Sig-KsRootSP}$）を生成して公開鍵証明書発行局（ＩＡ）１６０２に送信する。

【０１５０】公開鍵証明書発行局（ＩＡ）１６０２は、受信データE_{Ks}（｛ＳＡＭＩＤ，ＫｐＵＤ｝$_{Sig-KsRootSP}$）を復号して署名検証後、ユーザ２００１の公開鍵ＫｐＵＤに対応する公開鍵証明書の無効化解除処理を行なう。すなわち、先に図５を用いて説明したデータベース中のフラグを有効に設定する。さらに、公開鍵証明書発行局（ＩＡ）１６０２は、有効化処理が実行されたか否か（ＯＫまたはＮＧ）の応答に署名およびセッション鍵による暗号化を行なったデータE_{Ks}（｛ＯＫ／ＮＧ｝$_{Sig-KsIA}$）をルート登録局（ルートＲＡ）１６０１に送信する。ルート登録局（ルートＲＡ）は必要に応じて失効リストの差分配布を行なう。

【０１５１】これらの一連の公開鍵証明書の失効解除処理が終了すると、ユーザ２００１の公開鍵はルート登録局（ルートＲＡ）１６０１の管理するサービス下で使用可能となる。

【０１５２】図２４は、公開鍵証明書の削除処理を説明する図である。この場合、ルート登録局（ルートＲＡ）１６０１は、公開鍵証明書発行局（ＩＡ）１６０２に証明書削除要求を発行して、さらにルート登録局（ルートＲＡ）１６０１と公開鍵証明書発行局（ＩＡ）１６０２間での相互認証処理を実行する。

【０１５３】相互認証が成立すると、ルート登録局（ルートＲＡ）１６０１は、削除する公開鍵のユーザ２００１の識別子ＳＡＭＩＳＤと、公開鍵ＫｐＵＤにルート登録局（ルートＲＡ）１６０１の秘密鍵で署名を実行して相互認証時に生成したセッション鍵で暗号化してE_{Ks}（｛ＳＡＭＩＤ，ＫｐＵＤ｝$_{Sig-KsRootSP}$）を生成して公開鍵証明書発行局（ＩＡ）１６０２に送信する。

【０１５４】公開鍵証明書発行局（ＩＡ）１６０２は、受信データE_{Ks}（｛ＳＡＭＩＤ，ＫｐＵＤ｝$_{Sig-KsRootSP}$）を復号して署名検証後、ユーザ２００１の公開鍵ＫｐＵＤに対応する公開鍵証明書の削除処理を行なう。すなわち、先に図５を用いて説明したデータベースから、対応する公開鍵情報を削除する処理を行なう。さらに、公開鍵証明書発行局（ＩＡ）１６０２は、削除処理が実行されたか否か（ＯＫまたはＮＧ）の応答に署名およびセッション鍵による暗号化を行なったデータE_{Ks}（｛ＯＫ／ＮＧ｝$_{Sig-KsIA}$）をルート登録局（ルートＲＡ）１６０１に送信する。

【０１５５】これらの一連の公開鍵証明書の削除処理が終了すると、ユーザ２００１の公開鍵はルート登録局（ルートＲＡ）１６０１の管理するサービス下で使用不可能となる。

【０１５６】次に、上述のルート登録局（ルートＲＡ）と登録局（ＲＡ）との階層構成において、登録局（ＲＡ）をシステムホルダ（ＳＨ）として設定した構成例について説明する。

【０１５７】システムホルダ（ＳＨ）は、例えばインターネット上で展開するインターネットショップマーケットを主催、管理する機関、携帯電話の通信インフラを提供する機関、ケーブルテレビのケーブル使用を管理する機関、電子マネー・カード発行主体等によって構成される。すなわち、システムホルダは、様々なコンテンツ、サービスを提供可能とするコンテンツまたはサービスの流通インフラを提供、管理し、デバイスの管理を行なう機関として定義される。

【０１５８】図２５にシステムホルダ２５０１、コンテンツクリエイタ２５０２、サービスプロバイダ２５０３、ユーザ２５０４の関係図を示し、図２６にシステムホルダ、コンテンツクリエイタ、サービスプロバイダ、ユーザデバイスの具体例を示す。

【０１５９】図２５において、システムホルダ２５０１は、コンテンツクリエイタ２５０２および、サービスプロバイダ２５０３、ユーザ（デバイス）２５０４において利用可能なコンテンツまたはサービス流通インフラを提供する。コンテンツクリエイタ２５０２および、サービスプロバイダ２５０３は、システムホルダ２５０１の提供するインフラを利用してコンテンツの提供あるいはサービスの提供を行なう。ユーザ（デバイス）２５０４は、システムホルダ２５０１の提供するインフラを利用

してサービスプロバイダ２５０３の提供するサービスを受ける。

【０１６０】図２６に、具体的なシステムホルダ、コンテンツクリエイタ、サービスプロバイダ、ユーザデバイスの例を示す。図２６に示すように、例えば、システムホルダ（ＳＨ）が、インターネットショップマーケットを主催、管理する機関である場合、コンテンツクリエイタ（ＣＣ）は、インターネットショップマーケットに提供される商品を提供する。サービスプロバイダ（ＳＰ）は、提供された商品をインターネットショップにおいて販売するショップ（店）であり、ユーザデバイスは、インターネットショップを利用するＰＣ等である。

【０１６１】また、システムホルダ（ＳＨ）が、通信会社等、携帯電話通信インフラの提供機関である場合、コンテンツクリエイタ（ＣＣ）は、携帯電話の通信インフラを利用して提供可能なコンテンツ、商品を作成、製造する。サービスプロバイダ（ＳＰ）は、コンテンツクリエイタ（ＣＣ）から提供されるコンテンツ、商品を携帯電話の通信インフラを利用してユーザに対して販売、提供する。この場合のユーザデバイスは、携帯電話となる。

【０１６２】また、システムホルダ（ＳＨ）が、ケーブルテレビのケーブル通信管理会社等、ケーブルテレビ通信インフラの提供機関である場合、コンテンツクリエイタ（ＣＣ）は、ケーブルテレビの通信インフラを利用して提供可能なコンテンツ、商品を作成、製造する。ケーブルテレビに提供される番組もコンテンツに含まれる。サービスプロバイダ（ＳＰ）は、コンテンツクリエイタ（ＣＣ）から提供されるコンテンツ、商品をケーブルテレビの通信インフラを利用してユーザに対して販売、提供する、例えば視聴者から直接、視聴料金を徴収するケーブルテレビ会社等である。

【０１６３】また、システムホルダ（ＳＨ）が、電子マネーの発行機関等、電子マネー決済処理インフラの提供機関である場合、コンテンツクリエイタ（ＣＣ）は、電子マネーによって利用（購入）可能な商品を提供するコンテンツ、商品の提供機関であり、サービスプロバイダ（ＳＰ）は、コンテンツクリエイタ（ＣＣ）から提供されるコンテンツ、商品を電子マネーを利用可能なショップとして実現した販売店となる。この場合のユーザデバイスは、電子マネーを入力可能なＩＣカード等になる。

【０１６４】この他にも、様々なタイプのシステムホルダ（ＳＨ）があり、システムホルダに応じてコンテンツクリエイタ（ＣＣ）、サービスプロバイダ（ＳＰ）、ユーザデバイスが構成される。すなわち、システムホルダ（ＳＨ）は、コンテンツクリエイタ（ＣＣ）、サービスプロバイダ（ＳＰ）、ユーザデバイスによって利用可能なコンテンツ、サービスの提供を可能とするためのコンテンツまたはサービスの流通インフラを提供、管理する機関として定義される。

【０１６５】ここでは、前述の登録局（ＲＡ）の機能をシステムホルダ（ＳＨ）が担う構成とすることにより、ユーザにとって利用し易いコンテンツまたはサービスの流通構成について説明する。

【０１６６】まず、図２７を用いて、前述の登録局（ＲＡ）の機能をシステムホルダ（ＳＨ）に付与しない形態での公開鍵暗号方式によるコンテンツまたはサービスの流通構成について説明する。

【０１６７】図２７に示すように、ユーザが利用可能なサービスは様々、存在するが、各々が独自の公開鍵暗号方式、すなわち独自の審査、独自の登録により特定のサービスにおいてのみ有効な独自の公開鍵証明書を発行して特定サービスの提供を行なっている。この従来型のサービス提供構成を示したのが図２７である。図２７では、サービスＡを提供するグループ２７１０と、サービスＢを提供するグループ２７２０を示している。

【０１６８】サービスＡを提供するグループ２７１０には、サービスＡの提供のために利用可能な公開鍵証明書発行局（ＩＡ－Ａ）２７１１、公開鍵証明書の利用を要求するサービスプロバイダ（ＳＰ）２７１４、ユーザ（デバイス）２７１５の登録管理を実行する登録局（ＲＡ－Ａ）２７１２が設置され、登録局２７１２は、例えば公的な審査機関２７１３の審査に基づいて、サービスプロバイダ２７１４、ユーザ（デバイス）２７１５の登録を行ない、公開鍵証明書発行局（ＩＡ－Ａ）２７１１に証明書の発行を要求し、サービスプロバイダ２７１４、ユーザ（デバイス）２７１５の管理を行なう。なお、公開鍵証明書発行局（ＩＡ－Ａ）２７１１と登録局２７１２によって認証局Ａ（ＣＡ－Ａ）が構成される。

【０１６９】サービスＢを提供するグループ２７２０には、サービスＢの提供のために利用可能な公開鍵証明書発行局（ＩＡ－Ｂ）２７２１、公開鍵証明書の利用を要求するサービスプロバイダ（ＳＰ）２７２４、ユーザ（デバイス）２７２５の登録管理を実行する登録局（ＲＡ－Ｂ）２７２２が設置され、登録局２７２２は、例えば公的な審査機関２７２３の審査に基づいて、サービスプロバイダ２７２４、ユーザ（デバイス）２７２５の登録を行ない、公開鍵証明書発行局（ＩＡ－Ｂ）２７２１に証明書の発行を要求し、サービスプロバイダ２７１４、ユーザ（デバイス）２７２５の管理を行なう。なお、公開鍵証明書発行局（ＩＡ－Ｂ）２７２１と登録局２７２２によって認証局Ｂ（ＣＡ－Ｂ）が構成される。

【０１７０】このような構成において、例えばサービスＡの提供を受けるために、登録局（ＲＡ－Ａ）２７１２を介して登録を行い、サービスＡで適用可能な公開鍵証明書の発行を受けているユーザ２７１５が、サービスＢのサービスを受けようとした場合は、発行済みの公開鍵証明書は使用できない。ユーザ２７１５が、サービスＢのサービスを受けるためには、サービスＢを管轄する登録局（ＲＡ－Ｂ）２７２２を介して新たな登録手続きを

行なって新たな公開鍵証明書の発行を受けることが必須となる。

【０１７１】これを解決するには、図２７に示す公開鍵証明書発行局と登録局によって構成される認証局（ＣＡ）相互間で認証する構成としたり、あるいは認証局（ＣＡ）を階層構造とすることが考えられるが、認証局（ＣＡ）の処理負担の増加、認証局（ＣＡ）構造の複雑化を招くという欠点がある。一方、ユーザが複数のサービスを受けるためにサービス毎の複数の公開鍵証明書をデバイス中に格納する構成とすると、ユーザデバイスの記憶領域を公開鍵証明書の記憶のために多く使用することになる。このような構成は、例えばユーザデバイスがＩＣカードのような限定されたメモリ領域を有するデバイスにおいては問題である。

【０１７２】また、図２７のユーザデバイス２７１５とユーザデバイス２７２５との相互間で、例えばオフラインでの相互認証を行なおうとした場合、それぞれの管轄認証局（ＣＡ）が異なっているため認証処理が実行できないことになる。相互認証を有効に実行するためには、デバイス自身管轄の認証局の公開鍵と、相手デバイスの管轄の認証局の公開鍵の層法をデバイスに格納することが必要となり、様々な相手デバイスとの認証が必要となる場合には、格納公開鍵の数もさらに増加することになる。

【０１７３】このように、サービス毎に独立した管理を行なう図２７の構成では、様々な問題が発生する。この問題を解決するのが図２８に示すシステムホルダ（ＳＨ）をルート登録局（ルートＲＡ）の下の階層に設定した構成である。

【０１７４】図２８の構成について説明する。図２８の構成は、先の図２７の構成に対応した構成であり、図の左側がサービスＡ、右側がサービスＢを提供するサービスプロバイダ集合が含まれる。サービスプロバイダ２８０４は、サービスＡの提供主体であり、サービスプロバイダ２８０７は、サービスＢの提供主体である。

【０１７５】サービスプロバイダ２８０４、ユーザ（デバイス）２８０５、サービスプロバイダ２８０７、ユーザ（デバイス）２８０８が認証対象者、すなわち公開鍵暗号化方式によるデータ送受信を実行する主体となる。図２８では、２つのサービスＡ，Ｂについての構成を示しているが、サービスは一般に多数存在することができる。

【０１７６】システムホルダＡ，２８０３は、前述の登録局（ＲＡ）としての役割、機能を実行する。管轄下にある認証対象者のサービスプロバイダ２８０４、ユーザ（デバイス）２８０５は、システムホルダＡ，２８０３に対して、自己の使用する公開鍵に対応する公開鍵証明書の発行を要求する。システムホルダＢ，２８０６は、管轄下にある認証対象者のサービスプロバイダ２８０７、ユーザ（デバイス）２８０８からの公開鍵証明書の発行要求を受領する。

【０１７７】システムホルダＡ，２８０３、システムホルダＢ，２８０６は、各サービスにおける対象（サービスに参加するエンティティ、機器）を認証する。また、システムホルダＡ，２８０３、システムホルダＢ，２８０６は、各サービスにおける対象（サービスに参加するエンティティ、機器、ユーザ）の使用する公開鍵の公開鍵証明書発行要求を受領し、これをルート登録局（ルートＲＡ）２８０２を介して公開鍵証明書発行局（ＩＡ）２８０１に転送する。ルート登録局（ルートＲＡ）２８０２は、認証済みのシステムホルダＡ，２８０３、システムホルダＢ，２８０６からの公開鍵証明書発行要求を受理する。すなわち、ルート登録局（ルートＲＡ）２８０２が公開鍵証明書発行要求を受領するのは、ルート登録局（ルートＲＡ）２８０２によって認証されたシステムホルダＡ，２８０３、システムホルダＢ，２８０６からの要求である。

【０１７８】図２８において、サービスプロバイダ２８０４、サービスプロバイダ２８０７は、例えは音楽データ、画像データ、ゲームプログラム等のコンテンツ配信のサービス提供を実行するサービスプロバイダであり、例えば、先に図２６を用いて説明した各種のサービスを提供するサービス提供主体によって構成される。

【０１７９】システムホルダＡ，２８０３、システムホルダＢ，２８０６は、サービスプロバイダ２８０４、サービスプロバイダ２８０７の提供するサービスを実現するインフラを管理する機関であり、図２６を用いて説明したように、携帯電話通信インフラ提供者、電子マネー・カード発行機関等によって構成される。

【０１８０】本実施例の特徴は、コンテンツ提供、サービス提供を実現するインフラを提供または管理する機関であるシステムホルダが公開鍵証明書による認証、データ通信を実行するサービスプロバイダ、ユーザデバイスの公開鍵証明書発行手続き仲介、登録管理を行なう点である。システムホルダは、コンテンツ提供、サービス提供を実現するインフラを提供または管理する機関であるので、そのインフラを利用するユーザ、あるいはサービスプロバイダの管理を行なっている場合が多く、管理用のデータベースを備えている構成である場合が多い。このような管理データベースを利用して公開鍵証明書発行先の管理を併せて行なうことで効率的なユーザ、あるいはサービスプロバイダ管理が実行可能となる。

【０１８１】また、例えば新たな通信インフラが構築され、新たなシステムホルダが出現した場合に、その新規システムホルダを既存のルート登録局（ルートＲＡ）、公開鍵証明書発行局（ＩＡ）の管轄下に設定することで、容易に新規のインフラを利用した公開鍵証明書発行構成が実現され、新たなインフラを利用したサービスの提供がいち早く実現できる。

【０１８２】ユーザデバイスは、１つの公開鍵証明書を

格納するのみで、様々なサービスを利用可能となる。すなわち、図２８の構成では、１つのルート登録局（ルートＲＡ）、公開鍵証明書発行局（ＩＡ）が様々なシステムホルダ、サービスプロバイタに対応して設定されているので、ユーザデバイスは１つの公開鍵証明書を持つことにより、異なるサービスにおいて利用可能となる。また、異なるシステムホルダの管轄下のユーザデバイス相互間においても、１つの共通する公開鍵証明書発行局（ＩＡ）の発行する公開鍵を用いることにより、相互認証が可能となる。

【０１８３】次に、本発明の公開鍵証明書発行システムおよびデータ通信方法における公開鍵証明書の利用構成の具体例を図２９を用いて説明する。

【０１８４】図２９の構成は、公開鍵の管理および公開鍵証明書の発行を行なう公開鍵証明書発行局（ＩＡ）２９０１、エンティテイ、すなわち公開鍵、公開鍵証明書の発行要求を行なう認証対象に対する確認処理を実行するルート登録局（ルートＲＡ）２９０２、登録局（ＲＡ）としてのサービスプロバイダ２９０３、クリアリングセンタ２９０４、そしてユーザ端末２９０５からなる。

【０１８５】ルート登録局（ルートＲＡ）２９０２は、公開鍵証明書発行局（ＩＡ）２９０１から認証を受けており、ルート登録局（ルートＲＡ）２９０２の公開鍵、秘密鍵、公開鍵証明書を保有している。さらに、ルート登録局（ルートＲＡ）２９０２の公開鍵および公開鍵証明書発行局（ＩＡ）２９０１の公開鍵は、ルート登録局（ルートＲＡ）２９０２の管理下にある登録局（ＲＡ）としてのサービスプロバイダ２９０３、クリアリングセンタ（ペイメントＲＡ）２９０４、そしてユーザ端末２９０５に通知、または各機器に埋め込まれている。

【０１８６】サービスプロバイダ２９０３、クリアリングセンタ２９０４は、ルート登録局（ルートＲＡ）２９０２に識別子を登録し、公開鍵証明書発行局（ＩＡ）２９０１から発行された公開鍵証明書を得る（図中の２）の処理）。

【０１８７】ユーザ端末２９０５は、サービスプロバイダ２９０３からサービスを受けるためには、ユーザ端末２９０５のＳＡＭ（Secure Application Module）を介してサービスプロバイダ２９０３である登録局（ＲＡ）に機器識別子を送信して機器登録をする（図中の３）の処理）。

【０１８８】サービスプロバイダ２９０３である登録局（ＲＡ）は、サービス提供者（ｅｘ．図示しないサービスプロバイダ、ショップ等）に対して機器識別子によって識別されるユーザ端末がサービス利用可能か否かを確認した後、ルート登録局（ルートＲＡ）２９０２、公開鍵証明書発行局（ＩＡ）２９０１経由で証明書を発行する。ユーザ端末２９０５は、発行された公開鍵証明書をユーザ端末内のＳＡＭに格納（図中の４））する。これらの処理により、ユーザ端末２９０５は、図２９に示すシステムにおいて公開鍵を用いたデータ通信が可能になり、サービスの提供を受けることができる。

【０１８９】ユーザ端末２９０５は、クリアリングセンタ（ペイメントＲＡ）２９０４からサービスを受けるためには、ユーザ端末２９０５のＳＡＭ（Secure ApplicationModule）を介してサービスプロバイダ２９０３である登録局（ＲＡ）に機器識別子を送信して機器登録をする（図中の６）の処理）。

【０１９０】また、ユーザ端末２９０５が例えばコンテンツ利用料金をユーザ端末のＳＡＭにセットされる電子マネーを使用して支払処理をクリアリングセンタ２９０４を利用して実行しようとする場合、クリアリングセンタ２９０４に対してユーザ端末２９０５の識別子を登録する。

【０１９１】クリアリングセンタ２９０４は、銀行等の決済機関に対して与信等を行ない、ユーザ（支払者）の身元確認を行ない、ルート登録局（ルートＲＡ）２９０２、公開鍵証明書発行局（ＩＡ）２９０１経由で証明書を発行する。ユーザ端末２９０５は、発行された公開鍵証明書をユーザ端末内のＳＡＭに格納（図中の７））する。

【０１９２】ユーザ端末２９０５は、サービスプロバイダ２９０３の管理するサービスの提供を受ける場合には、サービスプロバイダ２９０３を介して受領した公開鍵証明書を用いる。また、クリアリングセンタ２９０４のサービス、すなわち支払処理を行なう場合は、クリアリングセンタ２９０４を介して得た公開鍵証明書を用いる。

【０１９３】なお、クリアリングセンタ２９０４が、サービスプロバイダ２９０３を介してユーザ端末に渡された公開鍵証明書をそのまま使用したい場合には、クリアリングセンタ２９０４を介した公開鍵証明書の生成処理を実行せず、クリアリングセンタ２９０４とルート登録局（ルートＲＡ）２９０２との間での処理により、すでに作成済みの公開鍵証明書をクリアリングセンタ２９０４での決済に有効な公開鍵証明書とすることも可能である。

【０１９４】本発明の公開鍵証明書発行システムおよびデータ通信方法では、従来、各々のサービスプロバイダが個々に行なっていた証明書取得処理をルート登録局（ルートＲＡ）に任せることが可能となり、また、例えばコンテンツ配信処理を行なうプロバイダがコンテンツ配信に伴う決済処理を行なうために実行する銀行等の金融機関に対する与信（ユーザの信用照会処理）を行なうことなく、ルートＲＡの管轄するクリアリングセンタ（ペイメントＲＡ）にその処理を任せることが可能となる。すなわち、新たに電子配信ビジネスを開始しようとするサービスプロバイダは、証明書の発行管理をルート登録局（ルートＲＡ）と公開鍵証明書発行局（ＩＡ）に

委託し、決済処理をルート登録局（ルートＲＡ）の管轄する別の登録局（ペイメントＲＡ）に委託することが可能となり、サービスプロバイダはユーザ管理業務を行なうのみでユーザの公開鍵証明書を使用したサービス提供が可能となる。

【０１９５】なお、ユーザ管理業務についてもルート登録局（ルートＲＡ）に管理を委託することも可能であり、サービスプロバイダとしての登録局（ＲＡ）は必要に応じてユーザ情報、失効情報等をルート登録局（ルートＲＡ）から受け取る構成とすることも可能である。

【０１９６】また、本発明の公開鍵証明書発行システムおよびデータ通信方法では、公開鍵証明書発行局（ＩＡ）は、証明書の発行、管理業務を行ない、証明書を使用するユーザの登録処理等のユーザ管理は、ルート登録局（ルートＲＡ）に委託する構成であり、サービスの内容に依存するユーザの確認業務を実行する必要がなくなる。また、失効リストの管理についてもルート登録局（ルートＲＡ）が処理する構成であり、公開鍵証明書発行局（ＩＡ）は、ルート登録局（ルートＲＡ）の要求に従って、証明書の有効、無効、削除処理を行なうのみとなる。

【０１９７】このように、本発明の公開鍵証明書発行システムおよびデータ通信方法では、公開鍵証明書発行局（ＩＡ）、ルート登録局（ルートＲＡ）、登録局（ＲＡ）の処理が切り分けられ、従来のシステムのようにサービス毎にユーザの確認、証明書の発行、登録、管理を一律に新規に構成することが要請されず、既存のデータを使用して必要な部分のみを構築することによって公開鍵および公開鍵証明書を使用した新たなサービスを開始することが可能となる。

【０１９８】このように、本発明の公開鍵証明書発行システムおよびデータ通信方法においては、公開鍵証明書発行局（ＩＡ）が公開鍵証明書の発行処理を実行し、ルート登録局（ルートＲＡ）が公開鍵証明書発行局（ＩＡ）が発行した公開鍵証明書を使用するユーザの管理を実行する。従って、公開鍵証明書発行局（ＩＡ）が発行した公開鍵証明書を複数のサービスプロバイダ（登録局（ＲＡ）または登録局（ＲＡ）の管理するサービスプロバイダ）が提供する様々なサービスにおいて共通に使用することが可能となり、新たなサービスを行なおうとするサービスプロバイダが認証局の機能を構築する必要がない。

【０１９９】また、公開鍵証明書発行局（ＩＡ）が発行する公開鍵証明書は、標準フォーマットに基づくものであるため、既存の認証局の発行した証明書と互換性があり、既存システムと本発明のシステムとを混在させることも可能である。

【０２００】以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【０２０１】

【発明の効果】上述したように、本発明の公開鍵系暗号を使用したデータ通信システムおよびデータ通信方法によれば、公開鍵証明書発行局（ＩＡ）、ルート登録局（ルートＲＡ）、登録局（ＲＡ）の処理が切り分けられ、従来のシステムのようにサービス毎にユーザの確認、証明書の発行、登録、管理を一律に新規に構成することが要請されず、既存の構築システムを使用するとともに、必要な部分のみを構築することによって公開鍵および公開鍵証明書を使用した新たなサービスを開始することが可能となり、従来システムにおける認証局の負荷を軽減することが可能となる。

【０２０２】また、本発明の公開鍵系暗号を使用したデータ通信システムおよびデータ通信方法によれば、登録局（ＲＡ）をコンテンツまたはサービスの提供を可能とするコンテンツまたはサービスの流通インフラを提供または管理する機関であるシステムホルダとして構成したことにより、異なるインフラ間においても、共通の公開鍵証明書を利用した認証、データ通信が実行可能となり、ユーザデバイスにおいて、異なるプロバイダの提供する様々なサービスの利用、あるいはユーザデバイス間での相互認証処理が共通の公開鍵証明書を用いて実行することが可能となる。

【図面の簡単な説明】

【図１】公開鍵証明書の例を示す図である。

【図２】本発明の公開鍵暗号を使用したデータ通信システムの概要を説明する図である。

【図３】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局と、ルート登録局と、認証対象者の処理の概要を説明する図（例１）である。

【図４】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局と、ルート登録局と、認証対象者の処理の概要を説明する図（例２）である。

【図５】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局の有するデータ構成を説明する図である。

【図６】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書の構成を説明する図（その１）である。

【図７】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書の構成を説明する図（その２）である。

【図８】本発明の公開鍵暗号を使用したデータ通信システムにおける登録局のデータベース内のデータ構成を説明する図である。

【図９】本発明の公開鍵暗号を使用したデータ通信シス

テムにおける失効リストの構成を説明する図（その１）である。

【図１０】本発明の公開鍵暗号を使用したデータ通信システムにおける失効リストの構成を説明する図（その２）である。

【図１１】本発明の公開鍵暗号を使用したデータ通信システムにおいて適用可能な署名生成処理について説明する図である。

【図１２】本発明の公開鍵暗号を使用したデータ通信システムにおいて適用可能な署名検証処理について説明する図である。

【図１３】本発明の共通鍵／対象鍵暗号を使用したデータ通信システムにおいて適用可能な相互認証処理について説明する図である。

【図１４】本発明の公開鍵暗号を使用したデータ通信システムにおいて適用可能な相互認証処理について説明する図である。

【図１５】本発明の公開鍵暗号を使用したデータ通信システムの処理において使用される用語を説明する図である。

【図１６】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局とルート登録局間での事前登録処理を説明する図である。

【図１７】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局とルート登録局および登録局間での処理を説明する図である。

【図１８】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局とルート登録局および登録局間でのオフライン処理を説明する図である。

【図１９】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局とルート登録局と登録局、およびユーザ間での処理を説明する図である。

【図２０】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局とルート登録局およびサービスプロバイダ間での鍵更新処理を説明する図である。

【図２１】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局とルート登録局およびサービスプロバイダ間での鍵更新処理を説明する図である。

【図２２】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局とルート登録局およびユーザ間での鍵失効処理を説明する図である。

【図２３】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局とルート登録局およびユーザ間での鍵失効解除処理を説明する図である。

【図２４】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局とルート登録局およびユーザ間での公開鍵証明書削除処理を説明する図である。

【図２５】本発明の公開鍵暗号を使用したデータ通信システムにおけるシステムホルダと他機関との関係について説明する図である。

【図２６】本発明の公開鍵暗号を使用したデータ通信システムにおけるシステムホルダと他機関の具体例を説明する図である。

【図２７】システムホルダをルート登録局に対する階層構造としない場合の公開鍵証明書利用例を説明する図である。

【図２８】システムホルダをルート登録局に対する階層構造とした場合の公開鍵証明書利用例を説明する図である。

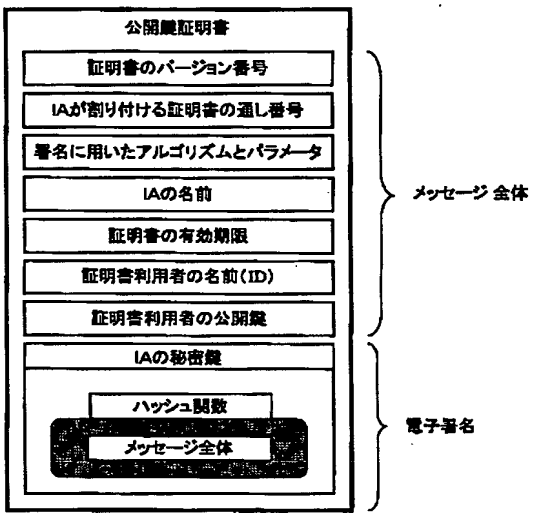
【図２９】本発明の公開鍵暗号を使用したデータ通信システムにおける公開鍵証明書発行局とルート登録局、登録局、ユーザ間での公開鍵証明書利用例を説明する図である。

【符号の説明】

２０１　公開鍵証明書発行局（ＩＡ）
２０２　ルート登録局（ルートＲＡ）
２０３，２０４　登録局（サービスプロバイダＲＡ）
２０５　登録局（ペイメントＲＡ）
２０６　ショップ
２０７　端末
２０８　ユーザデバイス
２０９　ユーザの決済機関
３０１　公開鍵証明書発行局（ＩＡ）
３０２　ルート登録局（ルートＲＡ）
３０３　認証対象者
１６０１　ルート登録局（ルートＲＡ）
１６０２　公開鍵証明書発行局（ＩＡ）
１７０１　登録局（ＲＡ）
２００１　ユーザ
２５０１　システムホルダ
２５０２コンテンツクリエイタ
２５０３　サービスプロバイダ
２５０４　ユーザ（デバイス）
２７１１，２７２１　公開鍵証明書発行局（ＩＡ）
２７１２，２７２２　ルート登録局（ルートＲＡ）
２７１３，２７２３　審査機関
２７１４，２７２４　サービスプロバイダ
２７１５，２７２５　ユーザ（デバイス）
２８０１　公開鍵証明書発行局（ＩＡ）
２８０２　ルート登録局（ルートＲＡ）
２８０３，２８０６　システムホルダ
２８０４，２８０７　サービスプロバイダ
２８０５，２８０８　ユーザ（デバイス）
２９０１　公開鍵証明書発行局（ＩＡ）
２９０２　ルート登録局（ルートＲＡ）
２９０３　登録局（サービスプロバイダＲＡ）
２９０４　登録局（ペイメントＲＡ）

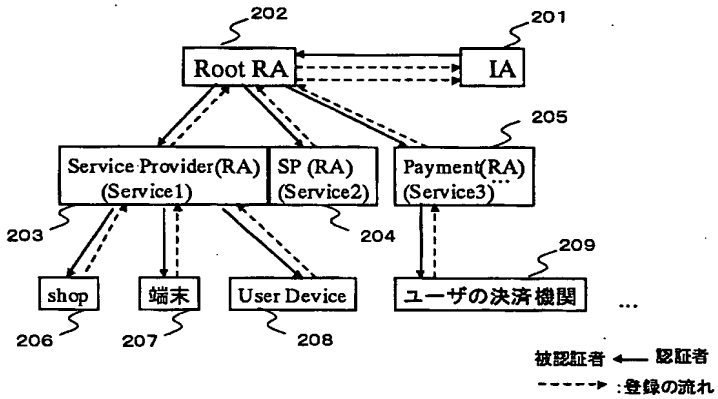
２９０５　ユーザ端末

【図１】

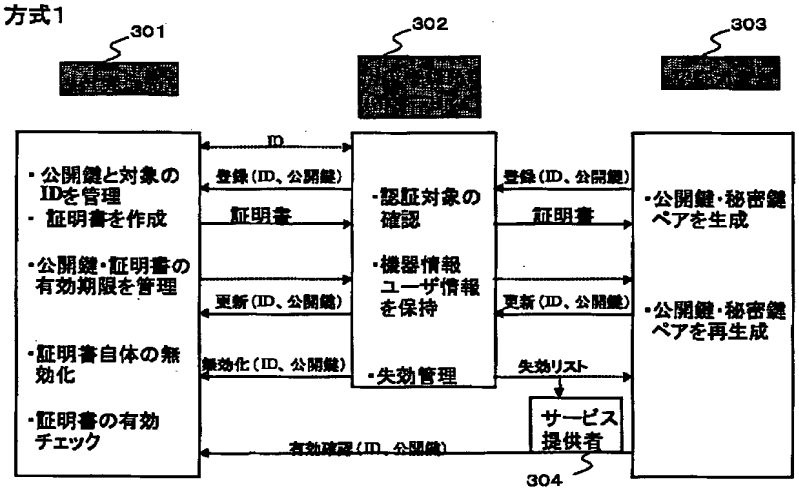


公開鍵証明書

【図２】



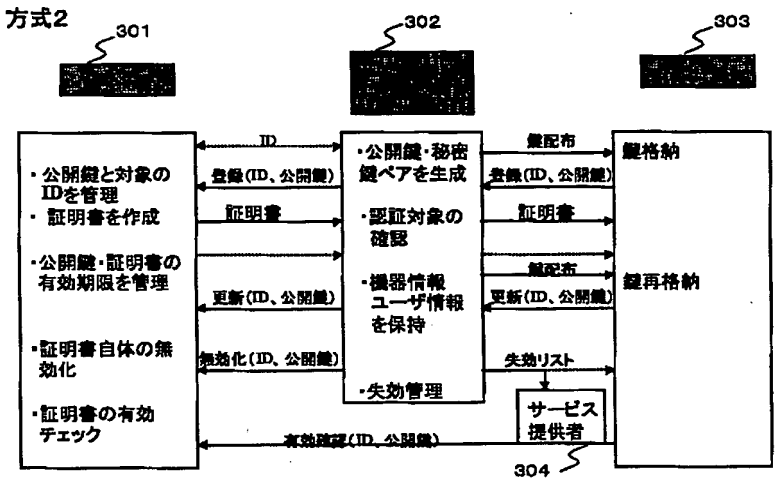
【図３】



【図５】

項目	内容
RA ID	サービス対象のRAのID
ID	対象を識別するID
公開鍵	対象の公開鍵
証明書	証明書本体
有効フラグ	有効か無効かを示すフラグ

【図４】



【図７】

policy Mappings issuer Domain Policy subject Domain Policy	CA を認証する場合にのみ必要。発行認証局のポリシーと被認証ポリシーのマッピングを規定	default＝なし
supported Algorithms algorithm Identifier intended Usage intended Certificate Policies	ディレクトリ (X.500) のアトリビュートを定義。コミュニケーションの相手がディレクトリ情報を利用する場合に、事前にそのアトリビュートを知らせるのに用いる。	default＝なし
subject Alt Name	user の別名 (GN 形式) 。	利用しない
issuer Alt Name	項目は入れておく(default＝なし)	default＝なし
subject Directory Attributes	userの任意の属性。	利用しない
basic Constraints cA path Len Constraint	証明対象の公開鍵が認証局の署名用か、user のものかを区別	default＝user用
name Constraints permitted Subtrees base minimum maximum excluded Subtrees	被認証者が認証局である場合 (CA 認証) にのみ使用。	default ＝ なし
policy Constraints requireExplicitPolicy inhibitPolicyMapping	認証パスの残りに対する明確な認証ポリシー ID、禁止ポリシーマップを要求する制限を記述	
CRL Distribution Points	user が証明書を利用する際に、証明書が失効していないかどうかを確認するための失効リストの参照ポイントを記述。	証明書を登録したところへのポインタ。失効リストは、発行元で管理
署名	発行者の署名	

【図６】

証明書フォーマット例（X.509 V3に準拠）

項目	説明	本IAにおける設定
Version1		
version	証明書のフォーマットのバージョン	V3
serial Number	IA によってつけられる証明書のSerial No.	シーケンシャルなシリアルナンバー
signature.algorithm Identifier algorithm parameters	証明書の署名アルゴリズム、及びそのパラメータ	楕円曲線暗号/RSA 楕円の場合パラメータ RSAの場合鍵長
issuer	IA 名 (Distinguished Name形式)	本ＩＡの名称
validity notBefore notAfter	証明書の有効期限 開始日時 終了日時	
subject	user を識別する名前	ユーザ機器ID またはサービス主体のID
subject Public Key Info algorithm subject Public key	user の公開鍵情報 鍵のアルゴリズム 鍵	楕円曲線/RSA user の公開鍵
Version3		
authority Key Identifier key Identifier authority Cert Issuer authority Cert Serial Number	IA の署名確認用の鍵識別 鍵識別番号 (8進数) IA名(General Name形式) 認証番号	
subject key Identifier	複数の鍵の証明をする場合	利用しない
key usage (0)digital Signature (1)non Repudiation (2)key Encipherment (3)data Encipherment (4)key Agreement (5)key CertSign (6)cRL Sign	鍵の使用目的を指定 (0)デジタル署名用 (1)否認防止用 (2)鍵の暗号化用 (3)メッセージの暗号化用 (4)共通鍵配送用 (5)認証の署名確認用 (6)失効リストの署名確認用	0,1,4,6 を利用
private Key Usage Period notBefore notAfter	user に格納されている秘密鍵の有効期限。	証明書の有効期限＝公開鍵の有効期限＝秘密鍵の有効期限 (default)

【図８】

エンティティDB

項目	内容
ID	対象を識別するID
認証データ	対象を認証するために必要な情報
認証結果	最新の認証結果(確認・与信結果など)
失効情報	以下の情報へのポインタ

【図９】

失効リストフォーマット例 (X.509 V2 に準拠)

・共通項目

項目	説明	
V1		
Signature.algorithm Identifier	署名アルゴリズム	楕円曲線暗号/RSA
Issuer	失効リスト発行局名	本ＩＡの名称
This Update	失効リストの発行日時	
Next Update	次の発行予定日	
V2		
Version	バージョン	
Authority key indentifier Key Identifier Authority Cert Issuer Authority Cert Serial Number	署名確認に用いるべき認証の識別子 鍵識別番号 (8 進数) IA 名(General Name形式) 認証番号	
CRL Number	失効リストの発行通し番号	
Issuing distribution point Distribution point Only contains user certs Only contaions CA certs Only some resons indirect CRL	失効リストの配布局の性質 配布局名(GN 形式) 加入者の失効専用の場合に「真」 CA 認証の失効専用の場合に「真」 いくつかの失効理由による 失効理由等の情報は失効リスト発行局ではなく認証発行局に迂回	
DeltaCRLIndicator	失効リストが差分かどうかの識別	

【図１５】

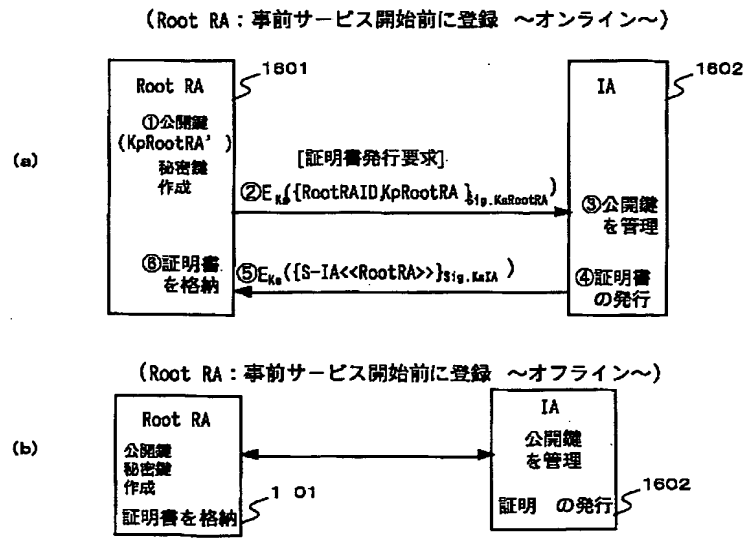
No.	用語	記号	説明・備考
1	公開鍵	K_{Pa}	Aの公開鍵。 例 User → K_{Pu} UD → K_{Pud} SP → K_{Psp} SB → K_{Psb}
2	秘密鍵	K_{Sa}	Aの秘密鍵。 例 User → K_{Su} UD → K_{Sud} SP → K_{Ssp} SB → K_{Ssb}
3	セッション鍵	Ks	相互認証の際、作成される共通鍵
4	証明書	A《B》	Aが発行したBの証明書。 例：IAによるUDの証明書 → IA《UD》
5	暗号化	E_{Ks}(data)	平文Dataを鍵Ksで暗号化
6	復号	D_{Ks}(data)	暗号文dataを鍵Ksで復号
7	署名	{data}Sig.K_{Sa}	DataをAの秘密鍵K_{Sa}で署名
8	署名付き暗号化	E_{Ks}({data}Sig.K_{Sa})	dataをAの秘密K_{Sa}で署名し，(data‖署名)を鍵Ksで暗号化

【図１０】

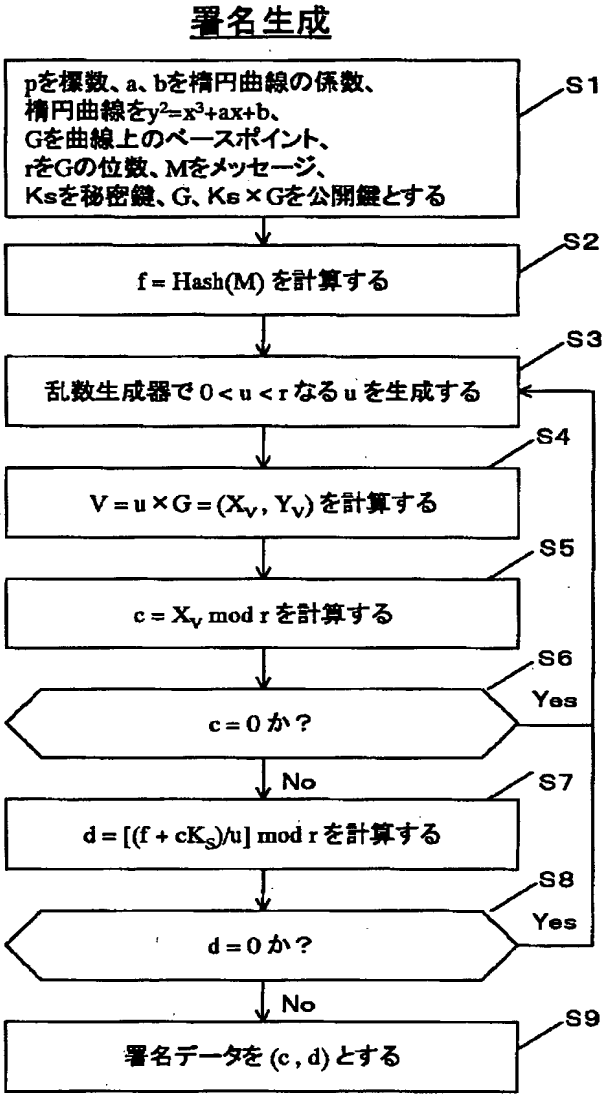
個別の証明書毎に管理される情報

項目	説明	
V1		
Certificate Serial Number	認証番号	
Revocation Date	失効申請受理日時	
V2		
Reason code	失効理由 0: 理由不明 1: 加入者の鍵が危瀕を受けた 2:CAの鍵が危瀕を受けた 3:認証の情報が変更 4:当該認証が置き換えられた 5:利用中止 6:利用の一時中止 7:一時中止の状態解除	
Hold instruction code	一時利用中止に対する対処方法	
Invalidity date	秘密鍵が被害にあったと考えられる日時	
Certificate issuer	認証発行局名(GN 形式) indirect 失効リストの場合は失効情報が、失効リスト発行局で管理されていないため、指定されたCAに迂回する。	
署名	発行者の署名	

【図１６】

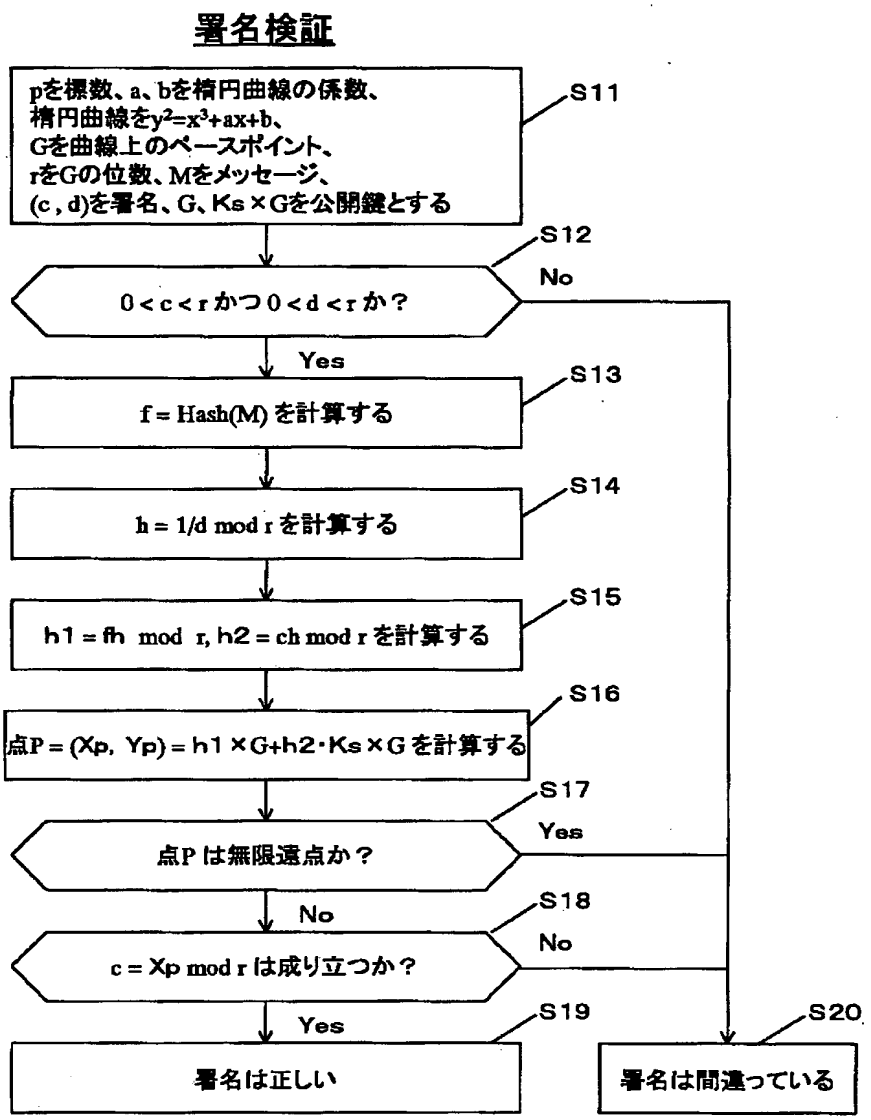


【図１１】



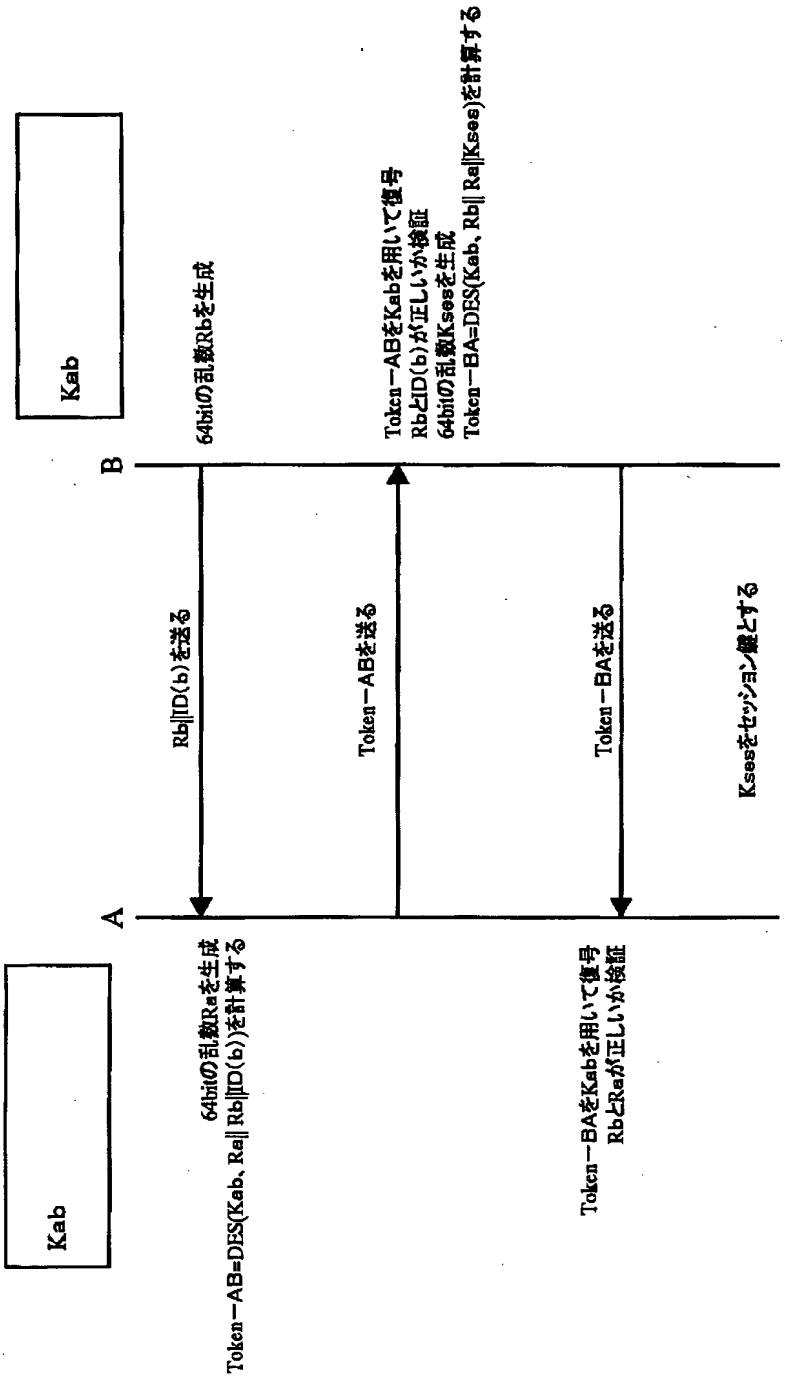
署名生成(IEEE P1363/D3)

【図１２】



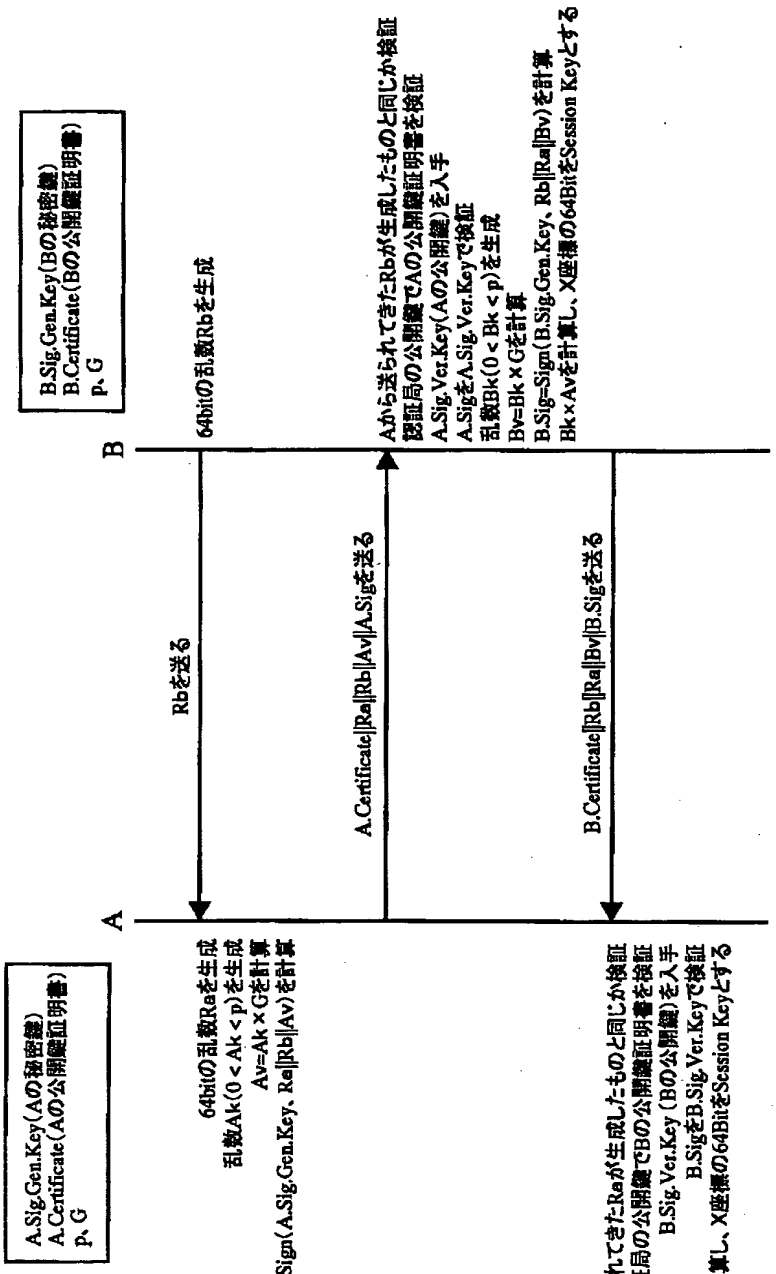
署名検証(IEEE P1363/D3)

【図１３】



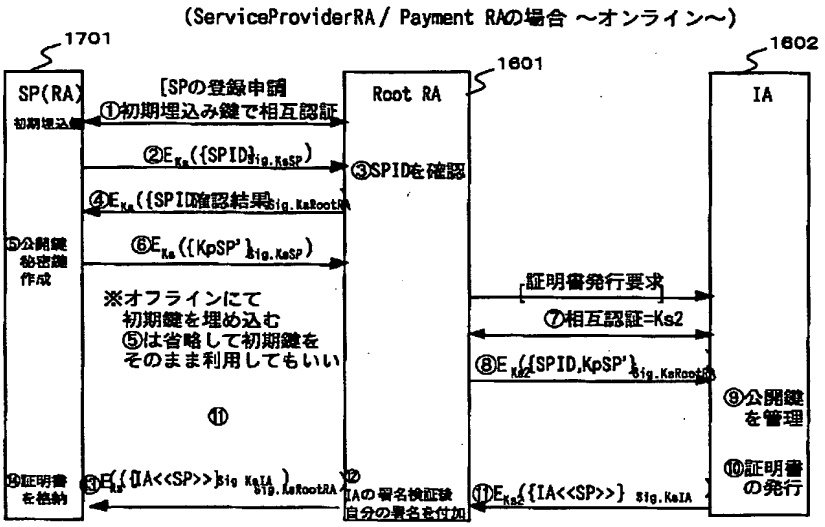
ISO/IEC 9798-2　対称鍵暗号技術を用いた相互認証および鍵共有方式

【図１４】

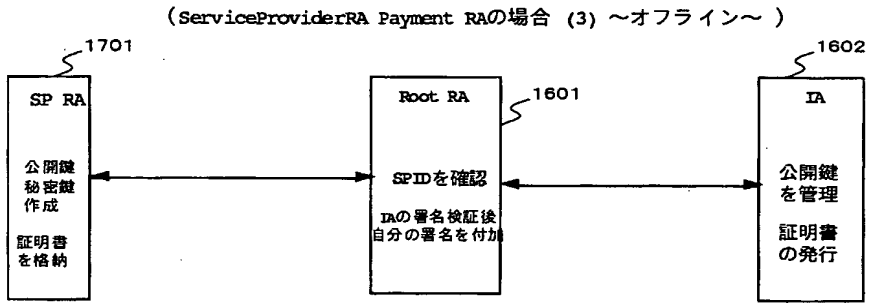


ISO/IEC 9798-3　非対称鍵暗号技術を用いた相互認証および鍵共有方式

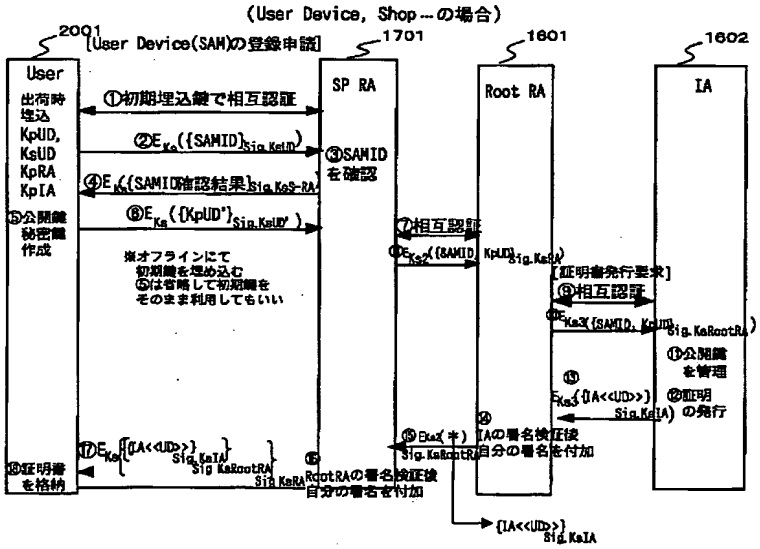
【図１７】


(ServiceProviderRA / Payment RAの場合 ～オンライン～)
1701
SP(RA)
初期埋込鍵
[SPの登録申請
①初期埋込み鍵で相互認証
②E_Ka({SPID}Sig.KsSP)
③SPIDを確認
④E_Ka({SPID確認結果}Sig.KaRootRA)
⑤公開鍵 秘密鍵 作成
⑥E_Ka({KpSP'}Sig.KsSP)
※オフラインにて
初期鍵を埋め込む
⑤は省略して初期鍵を
そのまま利用してもいい
⑪
⑨証明書 を格納
⑬E_Ks({IA<<SP>>}Sig.KsIA Sig.KaRootRA)
⑫ IAの署名検証後 自分の署名を付加
1601
Root RA
1602
IA
証明書発行要求
⑦相互認証=Ks2
⑧E_Ks2({SPID,KpSP'}Sig.KaRootRA)
⑨公開鍵 を管理
⑩証明書 の発行
⑪E_Ks2({IA<<SP>>}Sig.KsIA)

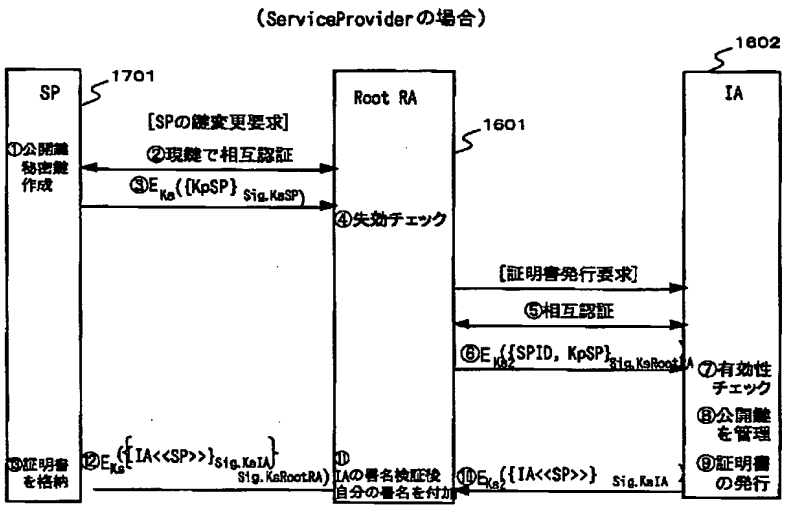
【図１８】


(ServiceProviderRA Payment RAの場合 (3) ～オフライン～)
1701
SP RA
公開鍵 秘密鍵 作成
証明書 を格納
Root RA
1601
SPIDを確認
IAの署名検証後 自分の署名を付加
1602
IA
公開鍵 を管理
証明書 の発行

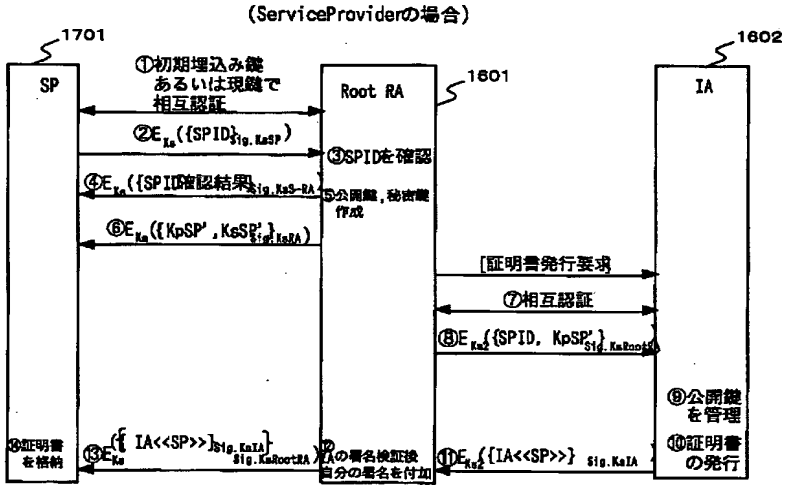
【図１９】


(User Device, Shop…の場合)
2001
[User Device(SAM)の登録申請]
1701
1601
1602
User
出荷時 埋込 KpUD, KsUD KpRA KpIA
SP RA
Root RA
IA
①初期埋込鍵で相互認証
②E_Ka({SAMID}Sig.KsUD)
③SAMID を確認
④E_Ka({SAMID確認結果}Sig.KaS-RA)
⑤公開鍵 秘密鍵 作成
⑥E_Ka({KpUD'}Sig.KsUD')
※オフラインにて
初期鍵を埋め込む
⑤は省略して初期鍵を
そのまま利用してもいい
⑦相互認証
⑧E_Ks2({SAMID, KpUD}Sig.KaSA)
[証明書発行要求]
⑨相互認証
⑩E_Ks3({SAMID, KpUD}Sig.KaRootRA)
⑪公開鍵 を管理
⑫E_Ks3({IA<<UID>>}Sig.KaIA)
⑫証明 の発行
⑬E_Ka({IA<<UID>>}Sig.KaIA Sig.KaRootRA Sig.KaRA)
⑭ IAの署名検証後 自分の署名を付加
⑮E_Ka(※)
Sig.KaRootRA
⑮RootRAの署名検証後 自分の署名を付加
⑰証明書 を格納
{IA<<UID>>}Sig.KsIA

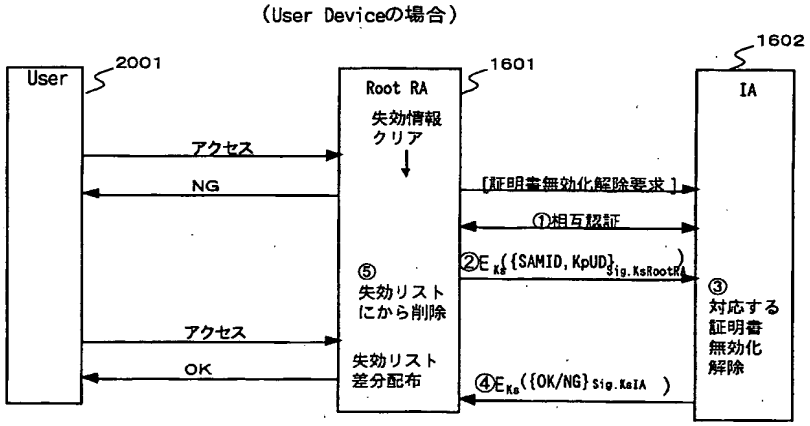
【図２０】


(ServiceProviderの場合)
SP
1701
Root RA
1601
IA
1602
[SPの鍵変更要求]
①公開鍵 秘密鍵 作成
②現鍵で相互認証
③E_Ks({KpSP}Sig.KsSP)
④失効チェック
[証明書発行要求]
⑤相互認証
⑥E_Ks2({SPID, KpSP}Sig.KsRootIA)
⑦有効性チェック
⑧公開鍵を管理
⑨証明書の発行
⑩E_Ks2({IA<<SP>>}Sig.KsIA)
⑪IAの署名検証後 自分の署名を付加
⑫E_Ks({IA<<SP>>}Sig.KsIA Sig.KsRootRA)
⑬証明書を格納

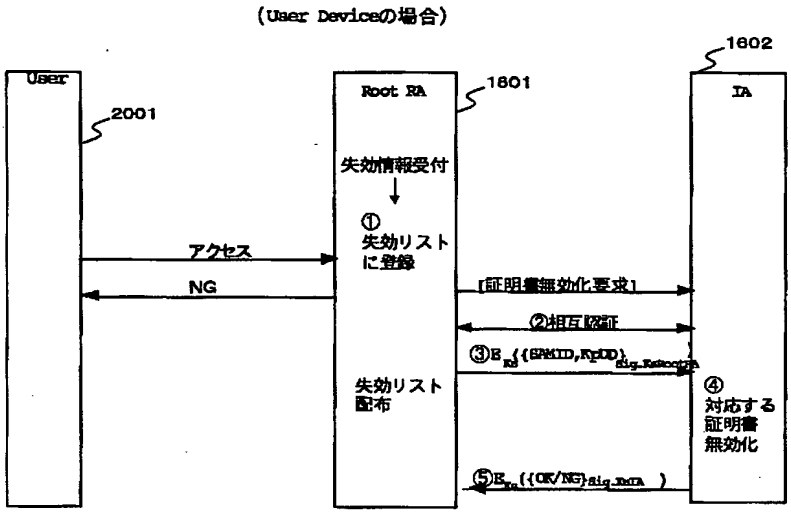
【図２１】


(ServiceProviderの場合)
SP
1701
Root RA
1601
IA
1602
①初期埋込み鍵 あるいは現鍵で 相互認証
②E_Ks({SPID}Sig.KsSP)
③SPIDを確認
④E_Ks({SPID確認結果}Sig.KsS-RA)
⑤公開鍵,秘密鍵 作成
⑥E_Ks({KpSP', KsSP'}Sig.KsRA)
[証明書発行要求]
⑦相互認証
⑧E_Ks2({SPID, KpSP'}Sig.KsRootRA)
⑨公開鍵を管理
⑩証明書の発行
⑪E_Ks2({IA<<SP>>}Sig.KsIA)
⑫IAの署名検証後 自分の署名を付加
⑬E_Ks({IA<<SP>>}Sig.KsIA Sig.KsRootRA)
⑭証明書を格納

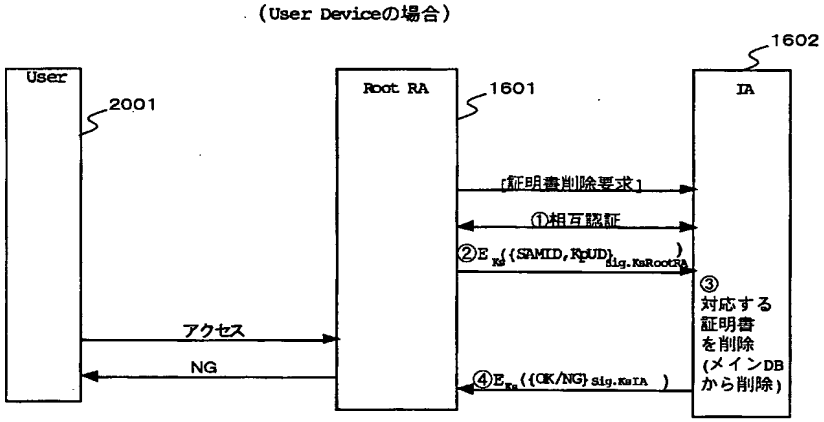
【図２３】


(User Deviceの場合)
User
2001
Root RA
1601
IA
1602
失効情報 クリア
アクセス
NG
[証明書無効化解除要求]
①相互認証
②E_Ks({SAMID, KpUD}Sig.KsRootRA)
③対応する証明書無効化解除
⑤失効リストにから削除
アクセス
OK
失効リスト差分配布
④E_Ks({OK/NG}Sig.KsIA)

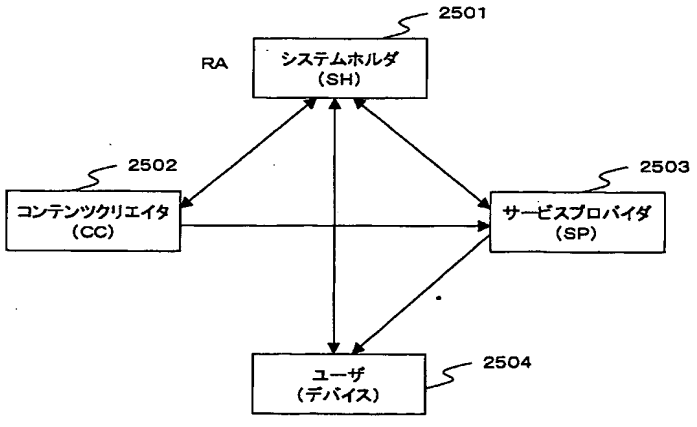
【図２２】


(User Deviceの場合)
User
2001
Root RA
1601
IA
1602
失効情報受付
①
失効リスト
に登録
アクセス
NG
[証明書無効化要求]
②相互認証
③E Ks{(SAMID,KpUD)sig.KsRootRA}
失効リスト
配布
④
対応する
証明書
無効化
⑤E Ks{(OK/NG)sig.KsIA}

【図２４】


(User Deviceの場合)
User
2001
Root RA
1601
IA
1602
[証明書削除要求]
①相互認証
②E Ks{(SAMID,KpUD)sig.KsRootRA}
③
対応する
証明書
を削除
(メインDB
から削除)
アクセス
NG
④E Ks{(OK/NG)sig.KsIA}

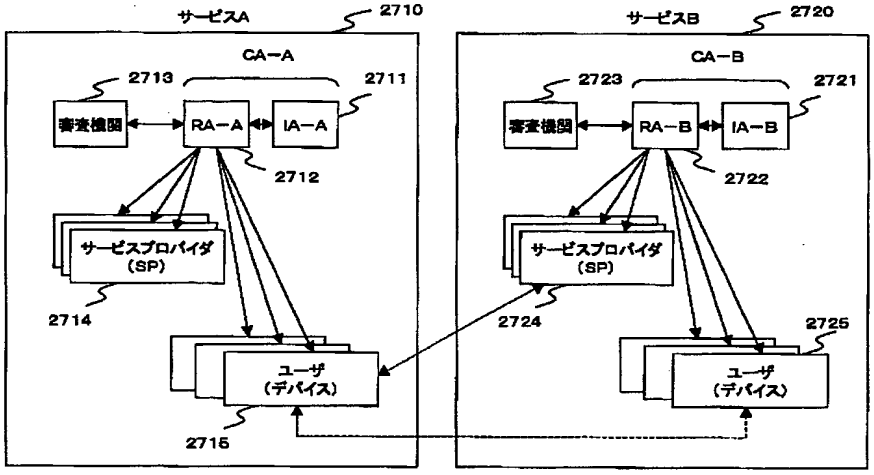
【図２５】


2501
RA
システムホルダ
(SH)
2502
コンテンツクリエイタ
(CC)
2503
サービスプロバイダ
(SP)
ユーザ
(デバイス)
2504

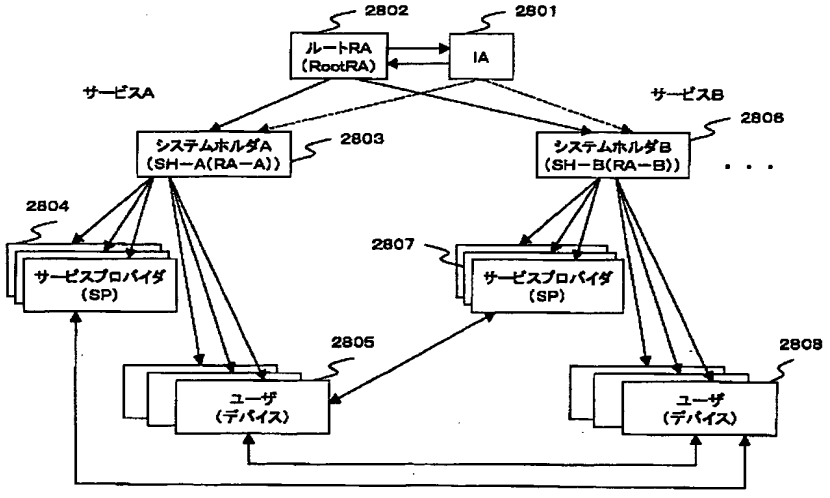
【図２６】

No.	システムホルダ(SH)	コンテンツ・クリエイタ(CC)	サービスプロバイダ(SP)	ユーザデバイス
1.	インターネットショップ マーケット主催機関	マーケット提供商品、 コンテンツの生成、製造者	マーケット提供商品ショップ	PC
2.	携帯電話通信インフラ 提供機関	携帯電話インフラを利用した 提供コンテンツ、商品の生成者	携帯電話利用ユーザに対する コンテンツ配信者	携帯電話
3.	ケーブルテレビ ケーブル管理機関	ケーブルTV番組制作者	ケーブルTV会社	TV(受像機)
4.	電子マネー・カード 発行機関	電子マネーにより購入可能な 商品、コンテンツの生成者	電子マネー利用可能ショップ	ICカード
5.	:	:	:	:
6.	:	:	:	:

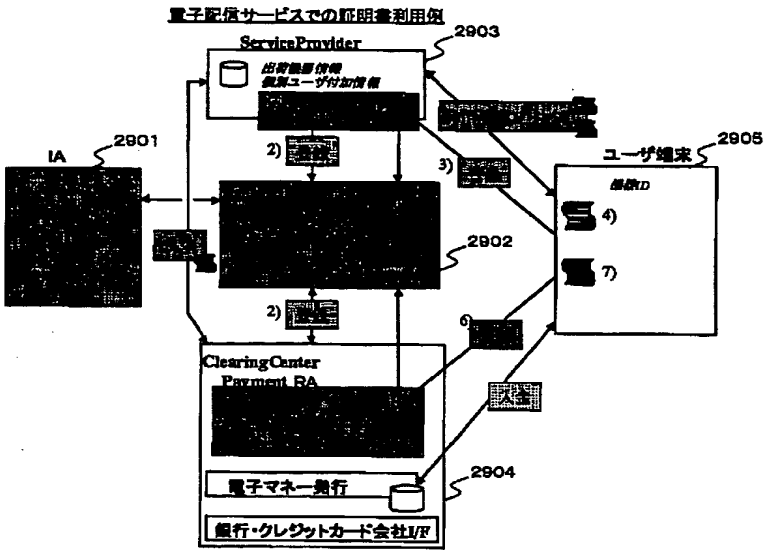
【図２７】



【図２８】



【図２９】



フロントページの続き

(72)発明者　松山　科子
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(72)発明者　昆　雅士
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(72)発明者　渡辺　秀明
東京都品川区北品川６丁目７番35号　ソニー株式会社内

Ｆターム(参考)　5J104 AA00 KA06 LA06 MA04 NA02
NA03 NA05 NA12 NA16 PA10